MIT/WHOI 2005-08

Massachusetts Institute of Technology
Woods Hole Oceanographic Institution

Joint Program in Oceanography/ Applied Ocean Science and Engineering

DOCTORAL DISSERTATION

Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles

by

Ryan M. Eustice

June 2005

MIT/WHOI

2005-08

Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles

by

Ryan M. Eustice

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

and

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

June 2005

DOCTORAL DISSERTATION

Funding was provided by the CenSSIS ERC of the National Science Foundation under grant EEC-9986821, in part by the Woods Hole Oceanographic Institution through a grant from the Penzance Foundation, and in part by a NDSEG Fellowship awarded through the Department of Defense.

This thesis should be cited as: Ryan M. Eustice, 2004. Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles. Ph.D. Thesis. MIT/WHOI, 2005-08.

Approved for publication; distribution unlimited.

Approved for Distribution:

W Rockwell

W. Rockwell Geyer, Chair
Department of Applied Ocean Physics and Engineering

Paola Malanotte-Rizzoli
MIT Director of Joint Program

John E Farrington

John W. Farrington
WHOI Dean of Graduate Studies

Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles

by

Ryan M. Eustice

B.S., Mechanical Engineering, Michigan State University (1998)

Submitted to the Joint Program in Applied Ocean Science & Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

June 2005

Author .... RM Eustice ....
Joint Program in Applied Ocean Science & Engineering
Massachusetts Institute of Technology
and Woods Hole Oceanographic Institution
April 29, 2005

Certified by .... Hanumant Singh ....
Hanumant Singh
Associate Scientist, WHOI
Thesis Supervisor

Certified by .... John J Leonard ....
John J. Leonard
Associate Professor, MIT
Thesis Co-Supervisor

Accepted by .... Mark Grosenbaugh ....
Mark Grosenbaugh
Chairman, Joint Committee for Applied Ocean Science & Engineering
Massachusetts Institute of Technology/
Woods Hole Oceanographic Institution

Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles

by

Ryan M. Eustice

Submitted to the Joint Program in Applied Ocean Science & Engineering
on April 29, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis describes a vision-based, large-area, simultaneous localization and mapping (SLAM) algorithm that respects the low-overlap imagery constraints typical of autonomous underwater vehicles (AUVs) while exploiting the inertial sensor information that is routinely available on such platforms. We adopt a systems-level approach exploiting the complementary aspects of inertial sensing and visual perception from a calibrated pose-instrumented platform. This systems-level strategy yields a robust solution to underwater imaging that overcomes many of the unique challenges of a marine environment (e.g., unstructured terrain, low-overlap imagery, moving light source).

Our large-area SLAM algorithm recursively incorporates relative-pose constraints using a view-based representation that exploits exact sparsity in the Gaussian canonical form. This sparsity allows for efficient $\mathcal{O}(n)$ update complexity in the number of images composing the view-based map by utilizing recent multilevel relaxation techniques. We show that our algorithmic formulation is inherently sparse unlike other feature-based canonical SLAM algorithms, which impose sparseness via pruning approximations. In particular, we investigate the sparsification methodology employed by sparse extended information filters (SEIFs) and offer new insight as to why, and how, its approximation can lead to inconsistencies in the estimated state errors. Lastly, we present a novel algorithm for efficiently extracting consistent marginal covariances useful for data association from the information matrix.

In summary, this thesis advances the current state-of-the-art in underwater visual navigation by demonstrating end-to-end automatic processing of the largest visually navigated dataset to date using data collected from a survey of the RMS Titanic (path length over 3 km and 3100 m^2 of mapped area). This accomplishment embodies the summed contributions of this thesis to several current SLAM research issues including scalability, 6 degree of freedom motion, unstructured environments, and visual perception.

Thesis Supervisor: Hanumant Singh
Title: Associate Scientist, WHOI

Thesis Co-Supervisor: John J. Leonard
Title: Associate Professor, MIT

Acknowledgments

Now that I'm just about to stop drinking from the fire hose, I realize both what a tremendous opportunity and experience I've had here in my n-plus years in the MIT/WHOI joint program (where it oftentimes seemed like n was closer to ∞ than to 0). The fact that I made it to this point is due to the help and influence of many others.

First, I'd like to thank my advisor, Hanu, who taught me that the answer to the ultimate question of life, the universe, and everything is 42, that all people are cordially referred to as "bubba", and that Pink Floyd's "Dark Side of the Moon" is the best album ever (actually I already knew that one, but it was reassuring to find out that my advisor was on the same wavelength). I also know that during my tenure here I was responsible for a number of gray hairs appearing on his head — to which I say both thank you and you're welcome. In addition, I'd like to acknowledge my appreciation for all of the opportunities and experiences he's given me (research cruises, conferences/workshops, India, SeaBED), as well as for helping me to see this through to the end.

Next, I'd like to thank my committee: John Leonard, Seth Teller, and Louis Whitcomb; they have provided valuable guidance and resources along the way. Specifically, to Louis for helping me carry out my validation experiments at the JHU test facility, to Seth for his meticulousness and for his help in defining my niche, and finally to my co-advisor John for imprinting upon me his useful idioms about graduate school: 1) "Must, Should, Would be nice", 2) "you're number one goal is to get out", and 3) "getting a PhD is about having a conversation with the literature."

Now that those formalities are out of the way, it's time to thank the people who actually toiled with me in the trenches, and from whom I probably learned the most from. To fellow advisees Chris Roman and Oscar Pizarro, what a strange long trip it has been — I think you'd be hard pressed to find three other guys that could have pulled it off better as a team than we did. I'm glad for all of the discussions about research and life, hanging out on movie nights with beer, and just generally make this an enjoyable experience. I'd also like to thank Matt Walter for his immense help in carrying out ideas on SEIFs (along with the cappuccinos), and the same goes for James Kinsey who helped me make the JHU experiment a success (the first round of Guinness is on me). Thanks also to Neil McPhee for his expertise in "making things work", to the scientists and staff of DSL for lending any help they could to students, to Marsha and Julia and everyone else in the Education Office for their amazing support, and to fellow cronies Mike Jakuba, Brian Bingham, Ali Can, Brendan Foley, and Rich Camilli for always making time to bounce ideas off of.

Finally, I'd like to thank my family for all of the love and support they have given me through the years. To my sisters Janae, Charla, and Jessy for always making me laugh, to my grandparents Gerald and Janet for being such an important part of my life, and to my mother Jeannie for inspiring me to dream big and see the world. Lastly, but definitely first, thank you to my 8 month old son, Noah, who is a source of tremendous joy in my life, and to my wife, Karen, who is my better nine-tenths and has provided support beyond words in helping me achieve this.

thank you all...
ryan

This work was funded in part by the CenSSIS ERC of the National Science Foundation under grant EEC-9986821, in part by the Woods Hole Oceanographic Institution through a grant from the Penzance Foundation, and in part by a NDSEG Fellowship awarded through the Department of Defense.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

ABE	Autonomous Benthic Explorer
AUV	autonomous underwater vehicle
BCC	brightness constancy constraint
CCD	charge coupled device
CI	covariance intersection
CG	conjugate gradients
CT	continuous-time
DOF	degree of freedom
DR	dead-reckoned
DSL	Deep Submergence Laboratory
DT	discrete-time
DVL	Doppler velocity log
ESDF	exactly sparse delayed-state filter
EM	electromagnetic
EKF	extended Kalman filter
EIF	extended information filter
FOG	fiber optic gyro
FOV	field of view
FFT	fast Fourier transform

GMRF	Gaussian Markov random field
GPS	global positioning system
IF	information filter
IFREMER	French Institute for the Research and Exploitation of the Sea
IFE	Institute for Exploration
INU	inertial navigation unit
INS	inertial navigation system
IR	infrared
JHU	Johns Hopkins University
KF	Kalman filter
LBL	long-baseline
LG	linear Gaussian
LMedS	least median of squares
MBARI	Monterey Bay Aquarium Research Institute
MBN	mosaic-based navigation
MIT	Massachusetts Institute of Technology
MLE	maximum likelihood estimate
MRF	Markov random field
NEES	normalized estimation error squared
PDE	partial differential equation
RANSAC	random sample consensus
RMS	Royal Mail Steamship
ROV	remotely operated vehicle
SEIF	sparse extended information filter
SIFT	scale invariant feature transform
SLAM	simultaneous localization and mapping
SNAME	The Society of Naval Architects and Marine Engineers
SSD	sum of squared differences

SFM	structure-from-motion
VAN	visually augmented navigation
USBL	ultra-short-baseline
WHOI	Woods Hole Oceanographic Institution

CHAPTER 1

Introduction

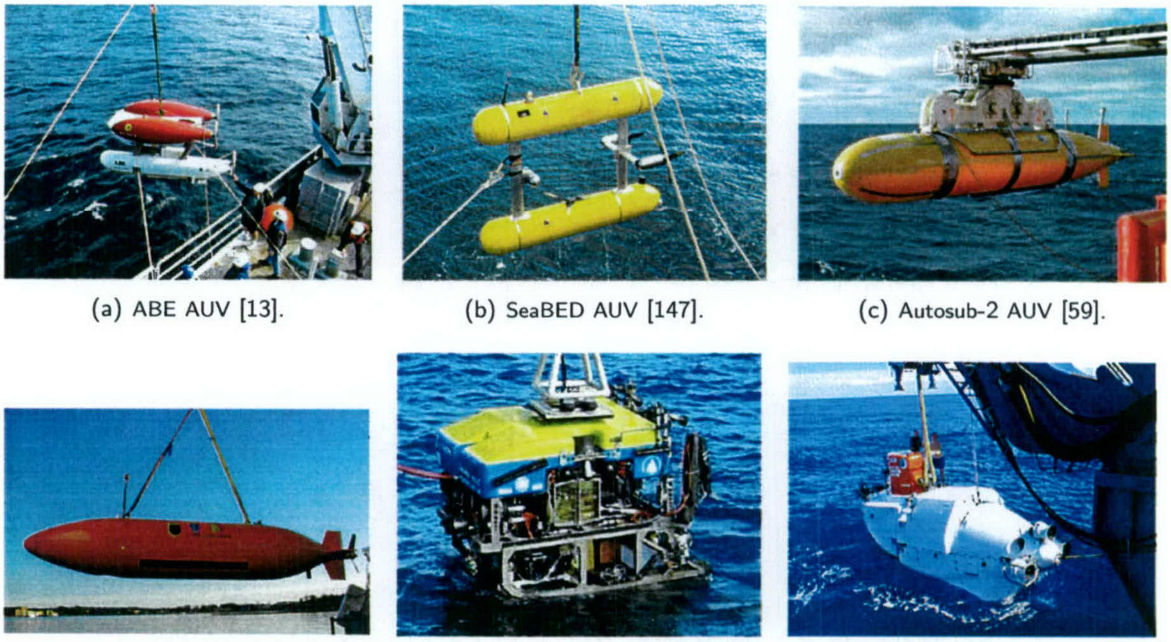
1.1 Motivation

Autonomous underwater vehicles (AUVs) are an emerging and enabling scientific technology that have seen significant advances and growth of user community over the past decade. A brief survey of recent literature shows the far-ranging impact AUVs have had in the research community. Applications include high-resolution geological mapping [153, 174–176] by ABE [13], coral reef habitat characterization [145, 146] by SeaBED [147], under-ice ocean exploration [15] by Autosub [59], and successful survey operations [165] by HUGIN [62]. See Fig. 1-1 for a depiction of the different vehicles. The growing popularity of AUVs arises from their unmanned and untethered design which makes them well suited to extended exploratory surveys requiring minimal user intervention and support. Meanwhile, their autonomous free-swimming capability has added a new paradigm of ocean sampling to the scientific user community as demonstrated by Fig. 1-2. They complement the capabilities of tethered remotely operated vehicles (ROVs) like Jason-2 [35] and free-swimming manned submersibles like Alvin [31, 129], both of which are well suited to intensive multi-sensor imaging and sampling of relatively small work areas.

The scientific user community has begun to embrace and exploit AUVs for their capacity to perform extended, exploratory, adaptive ocean sampling and mapping surveys. A précis of the diverse mission scenarios for which they are being deployed includes hydrothermal vents and spreading ridges [153, 174–176], chemical plume mapping [47, 66], studies of biodiversity [133], underwater forensics [69, 89, 148], deep-water archeology [5, 144, 178], and the monitoring of coral reefs [145]. Fig. 1-3 illustrates a few of these applications. For maximum utility, scientists typically require that AUVs be capable of georeferencing both the real-time survey and/or the collected data for post-processing. However, depending on the requisite spatial precision and desired survey extent, this requirement can pose a significant challenge due to the lack of easily obtainable large-area underwater precision navigation.

While the underwater realm presents its own peculiar challenges to autonomous navigation, the lack of easily obtainable precision navigation is by no means limited to the

Figure 1-1 A survey of state-of-the-art vehicle designs for deep-ocean science.



(a) ABE AUV [13].

(b) SeaBED AUV [147].

(c) Autosub-2 AUV [59].

(d) HUGIN AUV [62].

(e) Jason-2 ROV [35].

(f) Manned submersible Alvin [31].

sub-sea domain of robotics. In fact, the past decade of robotics literature shows that, in general, a fundamental issue of current interest in robotics is a solution to the joint task of simultaneous localization and mapping (SLAM) — a capability considered to be a key prerequisite of truly autonomous robots. SLAM represents a "chicken and egg" problem where the concept is deceptively simple. The robot's goal is to be able to autonomously navigate through an *a priori* unknown environment. To do this, it tries to navigate much like a person, by building a "mental map" of distinguishable landmarks in the environment that can be easily recognized when revisited. In this way, whenever the robot gets "lost" (i.e., accrues a lot of error in where it thinks it is), if it can sight one of its previously identified landmarks it can figure out where it is with respect to where it has been. The difficulty here is that the robot never quite knows its position *exactly* when building the map (due to accumulation of small errors while navigating) which is further complicated by the fact that its perceptual measurements of landmarks are never *perfect* (due to sensor inaccuracies). The net effect of these coupled errors is that the very map that the robot is trying to use to help improve navigation, inherently has distortions in it due to localization errors during its construction, which then affects its ability to navigate — hence, the simultaneous nature of the problem.

In essence, SLAM involves a joint-estimation problem over pose and map and has attracted a flurry of research over the past decade and a half since the seminal work by Moutarlier and Chatila [110] and Smith, Self, and Cheeseman [154, 155]. Since that time, significant advances have been made in dealing with several fundamental issues such as environmental scalability [11, 60, 85, 86, 108, 160] (i.e., how many landmarks can the robot maintain in its map), data association [87, 88, 118] (i.e., the problem of establishing land-

Figure 1-2 This figure shows the trajectories of a multi-phase deep-ocean survey by the ABE AUV [82]. Plotted on top of the corresponding bathymetric contour plot are the AUV navigation tracklines from a hydrothermal vent survey of the Lau Basin in the South Pacific. The different color tracklines represent multiple spatial resolution phases of the survey. As highlighted by the inset, each successive phase is conducted at a finer scale as ABE "homes in" on hydrothermal vent signatures. AUVs are especially suited for this task because they are both autonomous and free-swimming (i.e., unlike ROVs they are not constrained by human fatigue and ship tether restrictions).

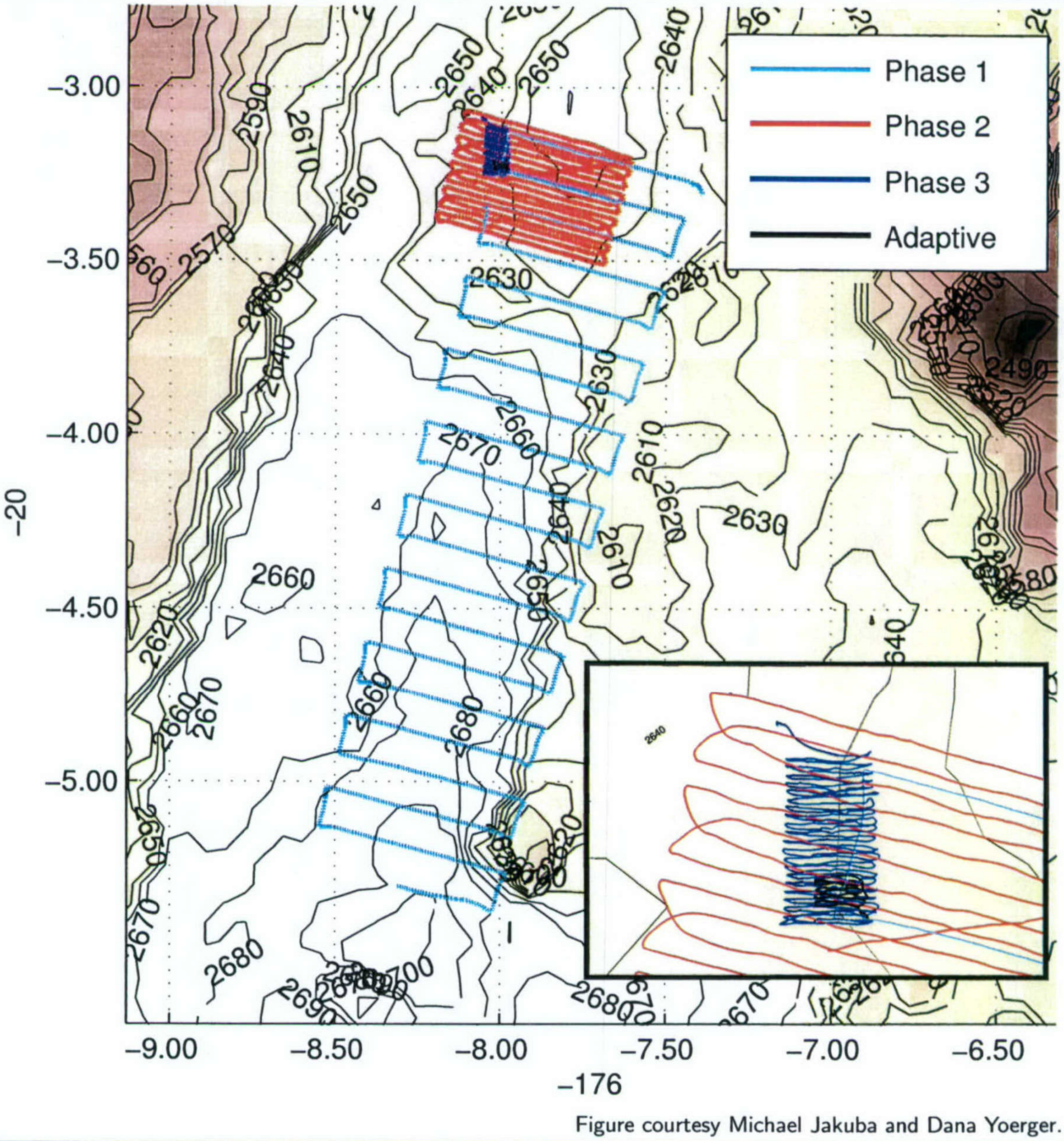
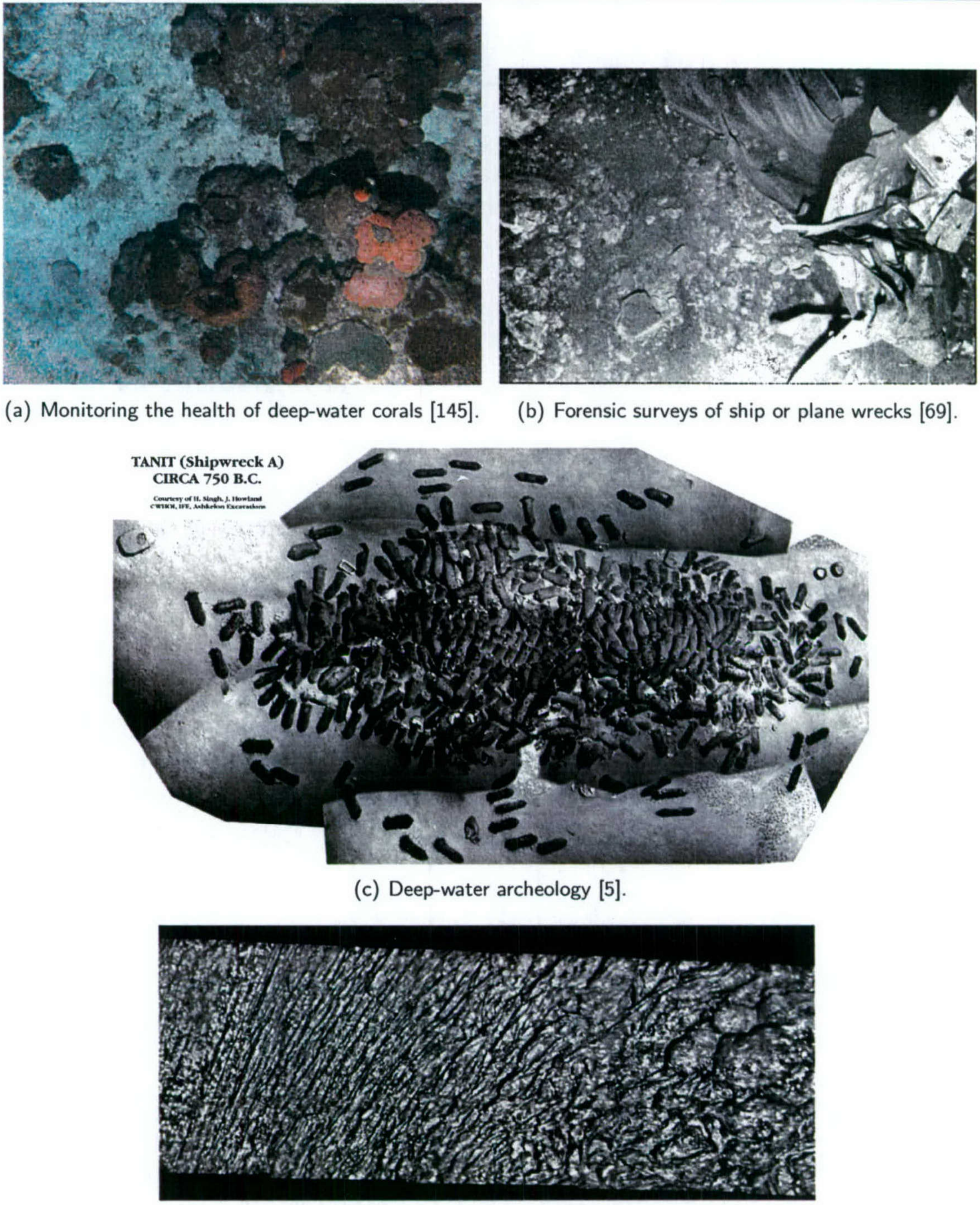


Figure courtesy Michael Jakuba and Dana Yoerger.

Figure 1-3 Some deep-water applications for AUVs.


TANIT (Shipwreck A)
CIRCA 750 B.C.
Courtesy of H. Singh, J. Howland
©WHOI, IFE, Ashkelon Excavations

(a) Monitoring the health of deep-water corals [145].

(b) Forensic surveys of ship or plane wrecks [69].

(c) Deep-water archeology [5].

(d) Geological surveys of spreading mid-ocean ridges [174].

mark correspondence to measurements), and map representation [94, 109, 154] (i.e., how to model the environment or landmarks within it). These advances have lead to the demonstration of impressive large-scale autonomous map making under challenging circumstances including large cyclic environments and poor odometry [11, 160, 161], and represent a significant fundamental achievement in our collective understanding of navigation with mobile robotics.

Thus, a SLAM framework seems like a natural choice for overcoming the current navigation limitations in the underwater domain so that we can better support near-seafloor ocean science. However, in trying to adopt a SLAM methodology for AUV navigation, a number of constraints quickly come to the fore. First, a large portion of the prior work in the SLAM literature has relied upon high-bandwidth, high-precision laser scanners as the perceptual sensor of choice for constructing accurate maps. Unfortunately, the strong attenuation of electromagnetic (EM) waves in the underwater realm generally limits our terrain sensing abilities to either an acoustic modality (frequency-dependent range and resolution) or near-field vision (1–5 m) [149]. Secondly, most mapped environments in the SLAM community are man-made, geometrically simple, indoor office spaces where a 2D map representation is sufficient and landmark features abound. However, in the underwater realm science drives the requirement that we must be able to navigate in 3D, rugged, unstructured, natural environments exercising full 6 degree of freedom (DOF) motion [144, 145, 153, 174–176, 178]. Hence, making SLAM a viable framework for improving AUV navigation requires general advances to overcome the particular constraints associated with a marine environment.

While the issues above pose significant challenges when employing SLAM in the underwater domain, AUVs themselves also offer some distinct advantages. For one, they tend to be well instrumented with advanced suites of inertial navigation sensors. This sensor suite may include a Doppler velocity log (DVL) [170] for measuring seafloor referenced velocities with mm/s precision and/or a North-seeking fiber optic gyro (FOG) [53] as a sub-degree heading reference (Table 1.1). Secondly, AUVs are typically used for studying benthic processes (e.g., hydrothermal vents and spreading ridges [174], deep-ocean corals [146], gas-blowout structures off the coastal shelf [66]) and as such, they routinely collect overlapping imagery of the seafloor using high-dynamic range CCDs. This implies that we can expect near-bottom AUVs to be equipped with a calibrated camera system in addition to other swath sensors [125] — see Fig. 1-4 for an illustration. Hence, our approach for overcoming near-seafloor navigation limitations has been to embrace a SLAM framework while explicitly exploiting the available sensor suite and rich calibrated visual imagery that is routinely collected during benthic underwater surveys.

1.2 A Review of Underwater Computer Vision

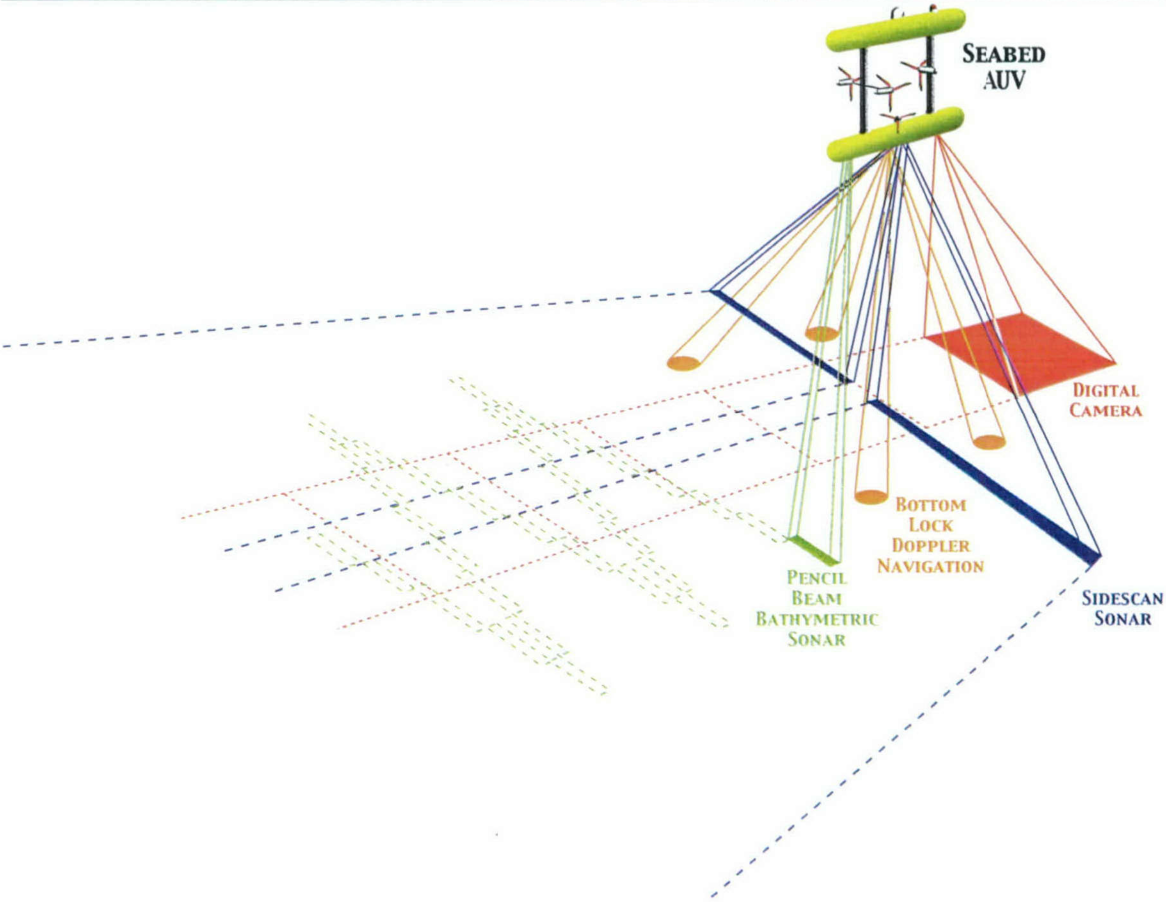
Computer vision is a broad research field that encompasses a diverse range of theory and application. Here, the discussion is restricted to areas that are essential to the understanding of underwater real-time visually-based navigation including aspects of image registration, mosaicking, epipolar geometry, and constraints peculiar to the marine environment.

Table 1.1 Current off-the-shelf underwater navigation sensors and systems.

INSTRUMENT	VARIABLE	INTERNAL?	UPDATE RATE	PRECISION	RANGE	DRIFT
Acoustic Altimeter	Z – Altitude	yes	varies: 0.1–10 Hz	0.01–1.0 m	varies	—
Pressure Sensor	Z – Depth	yes	medium: 1 Hz	0.01%	full-ocean	—
Inclinometer	Roll & Pitch	yes	fast: 1–10 Hz	0.1–1°	±45°	—
Magnetic Compass	Heading	yes	medium: 1–2 Hz	1–10°	360°	—
Gyro Compass	Heading	yes	fast: 1–10 Hz	0.1°	360°	10°/h
Ring-Laser Gyro	Heading	yes	fast: 1–1000 Hz	0.0018°	360°	0.44°/h
Bottom-Lock Doppler	XYZ – Velocity	yes	fast: 1–5 Hz	0.2–1.0%	30–200 m	—
12 kHz LBL	XYZ – Position	NO	varies: 0.1–1.0 Hz	0.01–10 m	5–10 km	—
300 kHz LBL	XYZ – Position	NO	1.0–5.0 Hz	±0.002 m typical	100 m	—

Adapted from Whitcomb [172] and Singh [151].

Figure 1-4 An illustration of the different types of swath sensors typically available on large AUVs.



1.2.1 Image Registration

Image registration is a fundamental task in computer vision, both at the micro level (e.g., pairwise registration) and the macro level (e.g., photogrammetry and large-scale bundle adjustment). Its objective is to relate two or more views of the same scene taken, for example, at different times, from different modalities (e.g., optical and infrared), or from different viewpoints. This task has application across many different disciplines spanning real-time target recognition and tracking [162], matching image pairs for recovering camera ego-motion and scene structure [157], aligning images from different modalities in medical diagnosis [91], and quantifying scene change detection [81].

In its most general form, image registration involves determining a mapping between images both spatially and with respect to intensity [17]. Defining an image as a two-dimensional array $I(u, v)$ where the spatial indices (u, v) map to a respective intensity, then the mapping between images I_i and I_j can be expressed as

$$I_j(u, v) = g(I_i(f(u, v)))$$

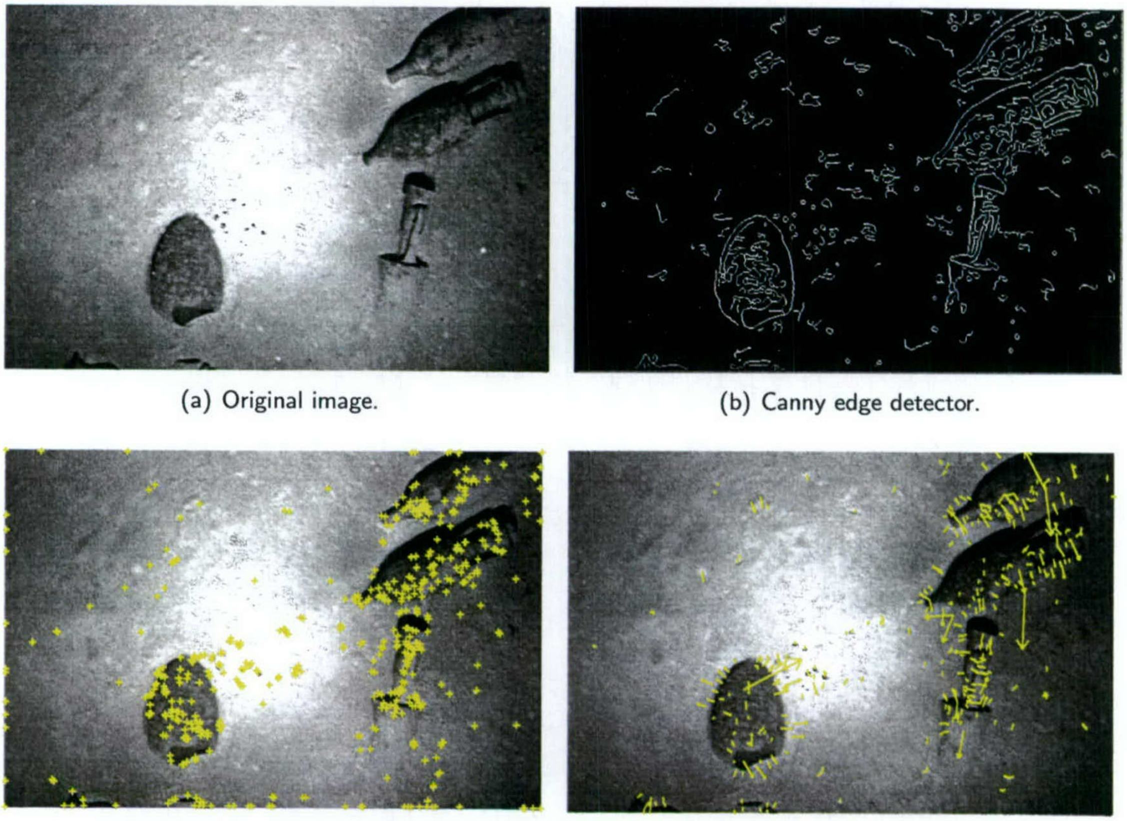
Here, $g(.)$ is a 1D intensity or radiometric transformation while $f(.)$ is a 2D spatial transformation mapping coordinates (u, v) to new coordinates $(u', v') = f(u, v)$. Generally speaking, the radiometric mapping $g(.)$ is explicitly considered when mapping from one modality to another where pixel intensities do not correspond to the same measurement (e.g., registering optical imagery to infrared imagery [72], or registering video imagery to a 3D-model depth map [166]). It is also relevant in imagery where the scene illumination varies and corresponding image points may not have the same intensity. For example, in deep-sea underwater imagery vehicles must carry their own illumination, resulting in varying scene brightness. Without loss of generality one can drop the explicit modeling of the radiometric mapping $g(.)$ and instead focus solely on the spatial registration $f(.)$ (i.e., $I_j(u, v) = I_i(f(u, v))$) noting that the radiometric mapping may either be considered as a preprocessing step [149], or incorporated into the spatial registration technique directly [72].

The spatial registration of images by the mapping $f(u, v)$ is generally based upon a *motion-model* [9, 80]. Two-dimensional global parametric motion-models are a useful class of mappings which can be applied across the whole image and are often used in mosaicking. These global transformations define a displacement $(\Delta u, \Delta v)$ for every pixel (u, v) in the image and range across increasing complexity from rigid, affine, projective, perspective, and polynomial transforms [17]. The different techniques used to determine the motion-model parameters can generally be divided into two classes — indirect methods and direct methods [17].

Indirect Methods

Indirect or feature-based methods generally rely upon condensing the large amount of image information into a small subset of feature tokens thereby reducing the amount of corresponding data to be matched. The first step in determining features is to apply some form of feature extraction to the image. This operation is generally desired to be invariant to a certain degree of image distortions, such as rotation and/or scaling, so that the same interest point may be picked out in both the reference and input images [140]. Operators include edges

Figure 1-5 This example illustrates some of the different types of features that can be extracted from underwater imagery using standard feature detectors.



(a) Original image.

(b) Canny edge detector.

(c) Harris corner detector.

(d) SIFT features.

and contours [22,90], corners [63], extremal regions [100], and scale-space maxima [93,106] as such operators extract the intrinsic local structure of an image. Examples of well-known operators, as illustrated in Fig. 1-5, include the Canny edge detector [22], the Harris corner detector [63], and most recently, the scale invariant feature transform (SIFT) [93]. The main goal of these operators is to pick out features of local interest in the image that contain information indicating the presence of easily distinguishable and meaningful characteristics in the scene.

Once features have been extracted, the next step is to establish their putative correspondence across overlapping imagery. Traditional methods establish feature correspondence by optimizing some type of local similarity metric such as correlation or equivalently sum of squared differences (SSD) [55]. While 2D correlation is computationally cheap as a similarity metric, it fails if feature regions differ by moderate rotations or scale differences. To overcome these limitations, more advanced techniques rely upon encoding some form of locally invariant feature descriptor. Differential invariants such as those described by Schmid [140], generalized image moments based upon Zernike polynomials [3,77,126], and Lowe's scale invariant feature transform [93] are all robust to scale and rotation. For more challenging registration problems, though, such as wide-baseline stereo applications, affine invariant feature descriptors have proven to be the most robust for establishing correspondences [106, 139, 164] (although they are also the most expensive computationally).

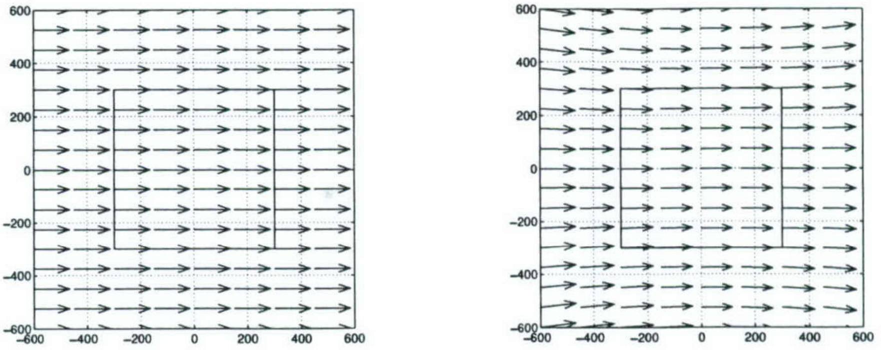
The last step in the registration process is to fit a parametric motion-model that describes the feature correspondences. Since the putative correspondence stage is often prone to error, robust outlier detection methods, such as least median of squares (LMedS) [134] or random sample consensus (RANSAC) [42], are typically employed in an iterative fashion to find a consistent inlier set and initial fit for the motion-model. Having done this, a maximum likelihood estimate (MLE) [64] is then typically performed as the final step.

Direct Methods

In contrast to feature-based indirect methods, direct methods work directly on the entire image to estimate the motion-model by computing measurable quantities from the raw pixel (intensity or color) values. A standard and well known technique has been the application of the brightness constancy constraint (BCC) [67], which assumes that an image point or small region corresponding to a particular scene point or surface patch remains approximately constant during the motion of the camera relative to the scene. This model is exact for Lambertian surfaces that are stationary with respect to the illumination source in the presence of a moving camera. The technique relies upon measuring the spatio-temporal image gradient (commonly referred to as *optical flow*) to estimate the image motion and, thus, generally requires video frame rates to satisfy the BCC's differential assumptions of spatial and temporal smoothness [112]. One of the weak points of this method for visual navigation is that it is susceptible to motion drift when integrated over time. This is particularly true for narrow field of view (FOV) cameras where the optical flow field is ambiguous for small translations parallel to the image plane versus small rotations along the pan and tilt axes (Fig. 1-6). While this method has been applied with good success in terrestrial applications where lighting is typically more uniform, it performs poorly when naïvely applied to underwater imagery due to a severe violation of the assumption that illumination be stationary with respect to the scene. Illumination of deep-sea imagery is necessarily time-varying, as vehicles have to carry their own light source. This results in varying scene irradiance across images, and in moving shadows. Negahdaripour [113, 117] has attempted to salvage the BCC, though, by incorporating a model for affine varying scene irradiance.

Other researchers have approached the problem of time-varying illumination by first transforming the images into an illumination-invariant representation [72, 149]. In [72] local normalized correlation is used in a pyramidal approach of maximizing normalized correlation surfaces to estimate the motion-model where the multi-resolution implementation allows for larger inter-image displacements [9]. This technique was originally developed for multi-sensor fusion of optical and infrared imagery and has been successfully applied underwater in at least one known structure-from-motion (SFM) application [97]. Meanwhile, in [149] a technique for underwater imagery is developed by first preprocessing using adaptive histogram specification [37, 182] (Fig. 1-7) followed by gray-level thresholding to detect and discount shadow regions in an attempt to account for lighting variations *before* attempting image registration. The effect of this preprocessing step is the masking of shadows and equalization of image contrast, which reduces the effects of lighting patterns.

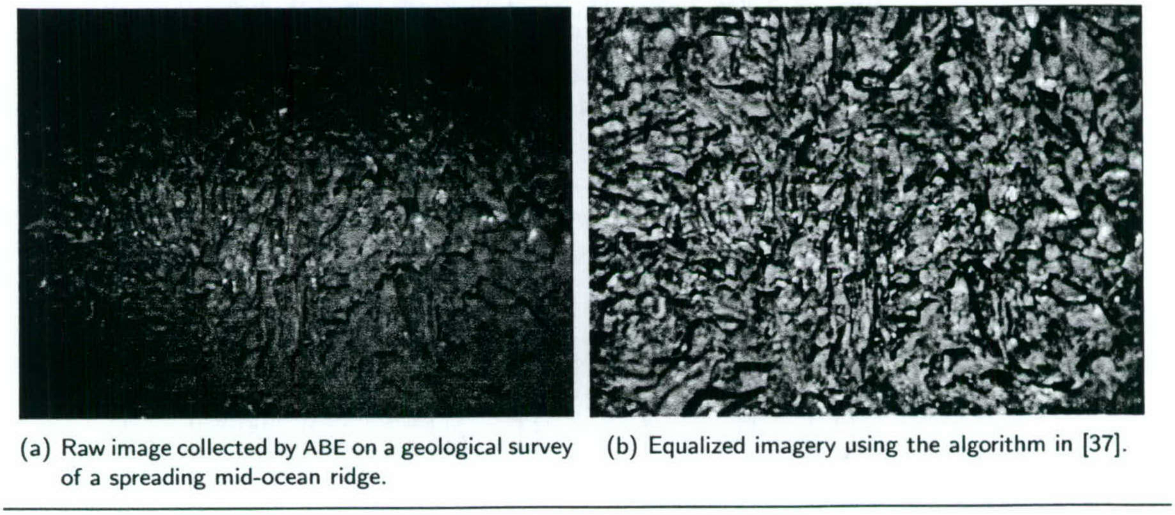
Figure 1-6 Optical flow methods for narrow FOV cameras suffer from a visual ambiguity between small camera translations versus small rotations. This simulated optical flow field was calculated for a small camera movement over a planar scene oriented parallel to the image plane. The two fields look nearly identical towards the center of the image (indicated by the black square). Note that most direct methods only process the central region of the image, both for efficiency and to minimize radial distortion effects.



(a) Small translation to the left.

(b) Small rotation (1°pan) to the left.

Figure 1-7 Varying scene illumination adds an additional challenge to underwater image registration. In this example the original image is preprocessed using contrast-limited adaptive histogram specification to compensate for vehicle lighting patterns before attempting to register imagery.



(a) Raw image collected by ABE on a geological survey of a spreading mid-ocean ridge.

(b) Equalized imagery using the algorithm in [37].

Correlation and its variants also fall under the umbrella of direct methods and are based upon finding the extrema of some form of signal similarity measure. From a computational standpoint, these techniques can be used when the images appear to be mostly displaced and, thus, have undergone little rotation or distortion so that a 2D translational search space is sufficient. Unfortunately, image distortions such as rotation and scaling are common in underwater imagery, as vehicle surveys are often unstructured or navigated with low resolution navigation [127]. In order to handle these higher-dimensional search spaces, correlation-based techniques can be posed more efficiently in the frequency domain by exploiting the phase shifting property of the Fourier transform to handle large rotations and scale changes.

The phase shifting property of the Fourier transform states that signals that are spatially shifted will result in transforms that are shifted in phase:

$$\begin{aligned} h_j(u,v) &= h_i(u-\Delta u, v-\Delta v) && \text{spatial domain} \\ H_j(\omega_1,\omega_2) &= H_i(\omega_1,\omega_2)e^{-j(\Delta u\omega_1+\Delta v\omega_2)} && \text{frequency domain} \end{aligned}$$

This property can easily be exploited to recover the unknown translation $(\Delta u, \Delta v)$ for 2D images and can also be extended to recover scale and rotation by representing these parameters in a coordinate system where they appear as shifts as described in [37,127,132]. Even more general affine motion-models can be represented by making use of additional Fourier properties [79,95]. These techniques benefit computationally by making use of the fast Fourier transform (FFT) and are insensitive to isolated frequency dependent image noise such as low-frequency illumination differences. Drawbacks, however, are: 1) they require a large overlap between image pairs to accommodate common area frequency representations, and 2) only linear motion-models can be described using Fourier transform properties. Fig. 1-8 demonstrates applying the frequency domain technique to register a pair of underwater images.

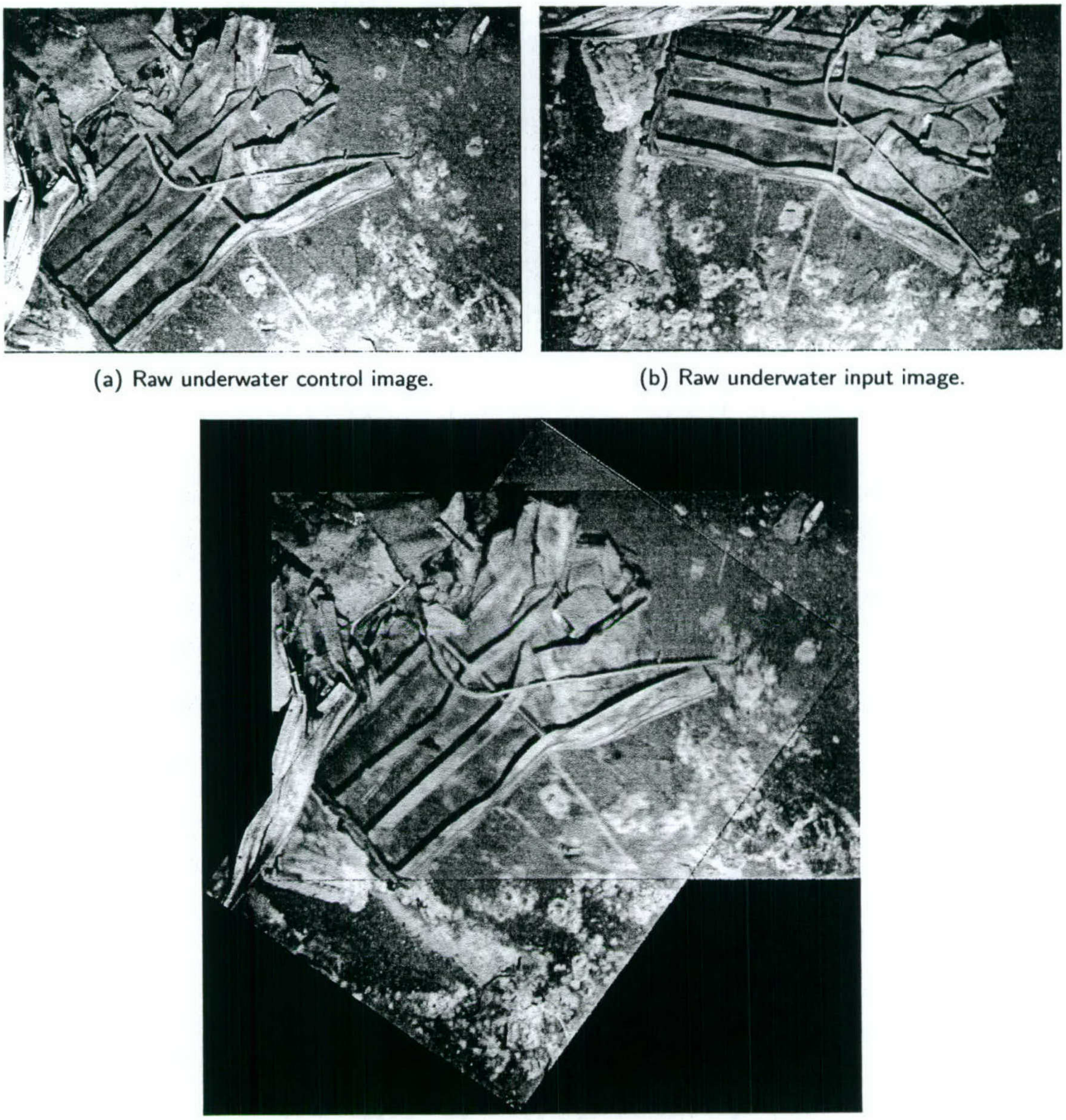
1.2.2 Mosaicking

Mosaicking is the task of combining two or more images such that the resulting composite image has an increased effective FOV. The problem has been extensively studied [73,80, 123,136,157], with early roots in aerial and satellite imaging where the planar parametric motion-model is well approximated due to the large separation between camera and scene. Planar parametric motion-models yield a composite image that is theoretically exact under only two conditions: 1) the scene structure is arbitrary and the camera undergoes rotation about its optical center, or 2) the camera motion is arbitrary, but the scene being viewed is planar [76]. Both of these conditions are equivalent to no observed parallax in the input images (Fig. 1-9).

Temporal Mosaicking

Early methods in mosaicking by the computer graphics community approached the problem in a temporally causal manner [73,80,123,136,157]. These approaches processed the imagery in a sequential manner to determine the pairwise homographies relating the temporal sequence, and constructed a composite view by concatenation (thus, warping all images to

Figure 1-8 This figure demonstrates using the 2D Fourier Transform technique to register a pair of underwater images collected during a forensic survey of the wreck of the M/V Derbyshire [69].

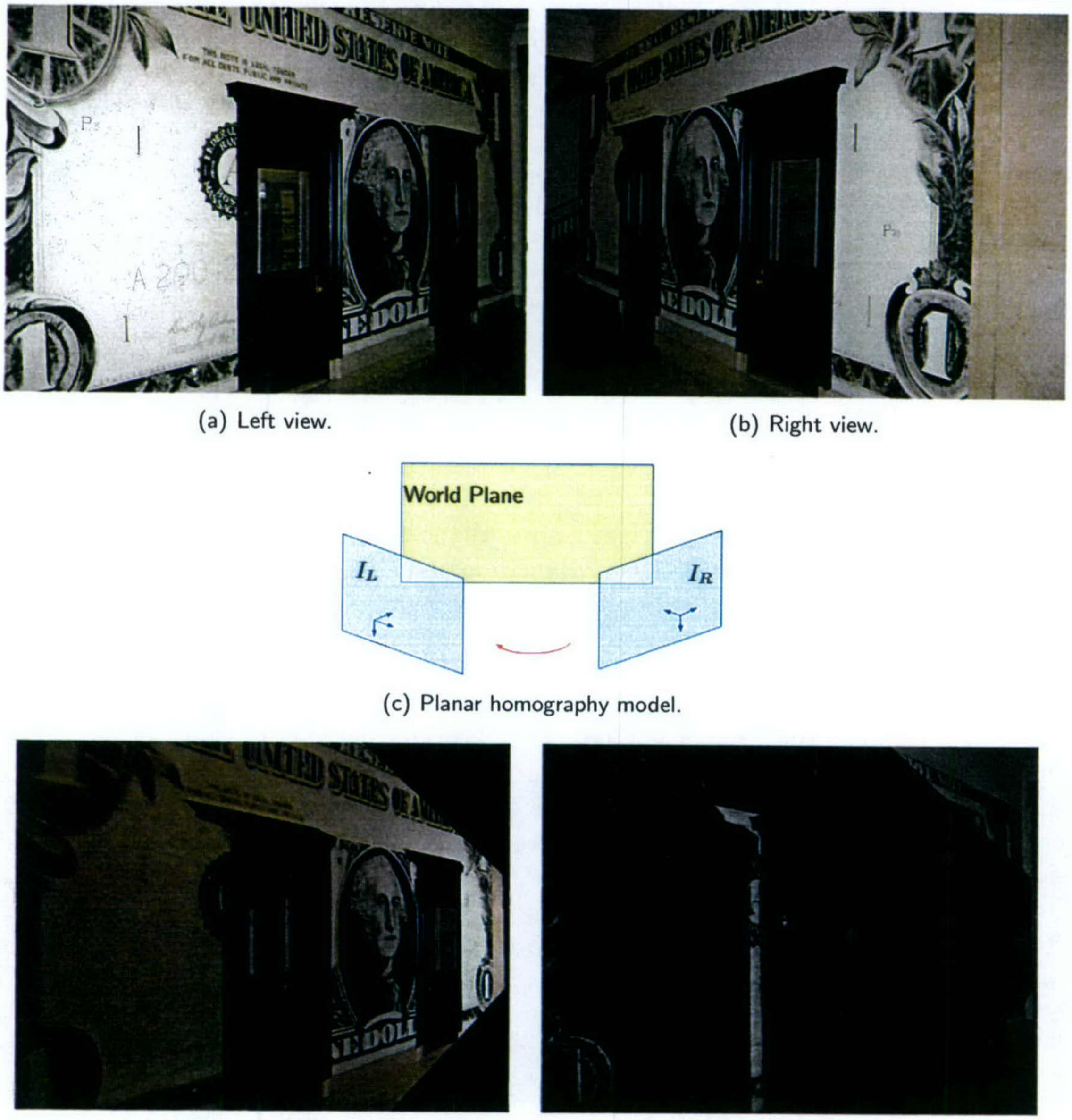


(a) Raw underwater control image.

(b) Raw underwater input image.

(c) Input image registered into the coordinate frame of the control image via Fourier methods [37] (average intensity is shown).

Figure 1-9 An example of scene-induced image parallax. Images (a) and (b), denoted I_L and I_R, correspond to two photos of a wall in front of the Cashier's Office at MIT taken from left and right vantage points respectively. Image (d), denoted I_W, is the result of warping I_R onto I_L using a planar perspective homography as illustrated in (c). Note that the area common to both (a) and (d) is in agreement except for the door jambs, which violate the planar scene assumption. Image (e) shows the pointwise difference between I_L and I_W. The discrepancy is due to parallax.



(a) Left view.

(b) Right view.

(c) Planar homography model.

(d) Warp of right to left.

(e) Their difference.

a common reference frame). While the pairwise homographies accurately describe the *local* registration, the small residual local alignment errors, coupled with errors in the applied motion-model, lead to an amplified *global* error when simply concatenated over long sequences. Since the image to reference frame homographies calculated by compounding do not attempt to achieve global consistency, images that are *not* temporal neighbors, but *are* spatial neighbors, may not be co-registered in the resulting mosaic.

Global Mosaicking

More recent efforts have focused on imposing the available non-temporal spatial constraints to produce a globally consistent mosaic [21, 126, 137, 138]. These methods formulate the problem as the optimization of a global cost function parameterized by all of the image to mosaic frame homography parameters. The mosaic topology may initially be derived in a coarse manner assuming simplified motion-model parameters between temporally connected neighbors. From this roughly estimated topology, new spatial neighbors are hypothesized and then tested. This process is iterated until a stable image topology emerges. The optimization of the cost function incorporates these spatial constraints to produce a globally consistent mosaic with enhanced quality and robustness as compared to simpler mosaicking methods.

1.2.3 Multiview Geometry

The Epipolar Constraint

While homographies can often be a useful approximation for obtaining composite views over an expanded FOV or for planar visual navigation [41], their shortcoming is that they can model only views of a single-plane environment. When the scene is nonplanar, more general descriptions of image motion must be employed. For example, the epipolar geometry for a pair of views is defined by the relative camera pose and allows for any ray from one image to be projected into the view of the other as demonstrated by Fig. 1-10 and Fig. 1-11. For a calibrated camera this geometric relationship is mathematically encoded in the *Essential matrix*, which is a 3×3 matrix that maps homogeneous normal coordinates from one image into the corresponding homogeneous epipolar line in the other. The epipolar line encodes for all possible scene depths the projection of a scene point into the view of the other camera. It can be used as an efficient 1D search constraint when trying to establish correspondences using a stereo camera setup where the relative camera pose is a known fixed quantity [29]. In the case of unknown calibration this relationship can be extended more generally through the *Fundamental matrix* [64, 179], which extends the concept of the Essential matrix by incorporating the unknown camera calibration into its definition and in recent years has become the focus of research due to the growing popularity of variable focus consumer digital-still and video cameras.

Structure from Motion

The most popular application of multiview geometry has been in that of offline structure-from-motion (SFM) [43, 96, 124]. SFM relates multiple views through either the Essential

Figure 1-10 Epipolar geometry model. The epipolar geometry is based upon ray projections between adjacent views and holds regardless of scene structure. In this example, the two scene points $\mathbf{X}_1$ and $\mathbf{X}_2$ are projected into scene points $\mathbf{x}_1$ and $\mathbf{x}_2$ and $\mathbf{x}'_1$ and $\mathbf{x}'_2$ in the left and right image planes using camera projection matrices P and P′ respectively. Note that each scene point in conjunction with the camera centers defines a plane, denoted the *epipolar plane*. The intersection of each epipolar plane with the image plane defines the *epipolar line*; it represents the projection of the corresponding scene ray as viewed by the other camera. When the geometry between the cameras (i.e., R and $\mathbf{t}$) is known, then the epipolar lines provide a 1D search constraint for establishing correspondences. Finally, note that the set of all epipolar planes defines a pencil whose intersection with the image planes defines the *epipoles* $\mathbf{e}$ and $\mathbf{e}'$. The line connecting the epipoles is called the *baseline* and corresponds to the vector $\mathbf{t}$.

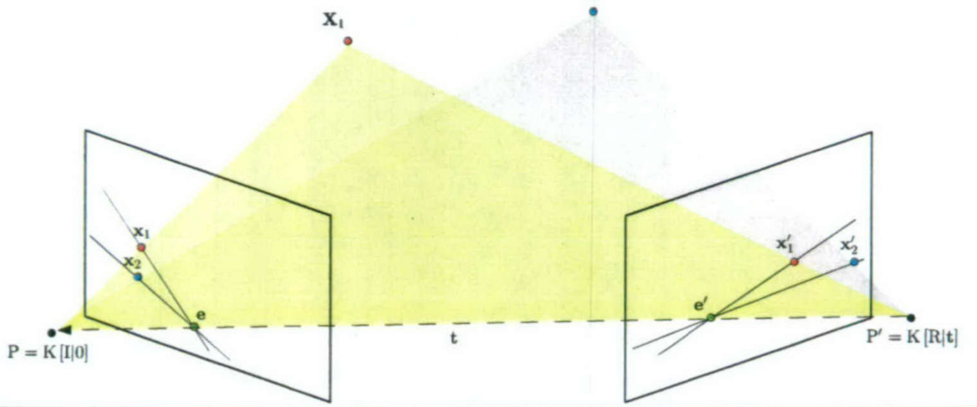


Figure 1-11 This figure demonstrates the epipolar geometry for a pair of successfully registered underwater images of the RMS Titanic. The epipolar lines are overlaid on the imagery and are color coded for correspondence (the circles along each line are the matched feature points). Note that the lines converge at the epipoles, which in this case are located outside of the viewable image.

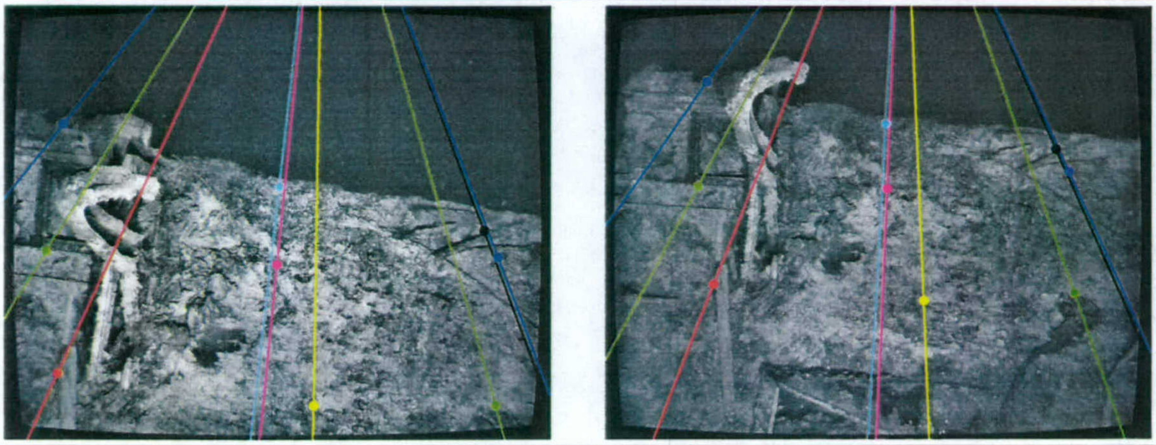
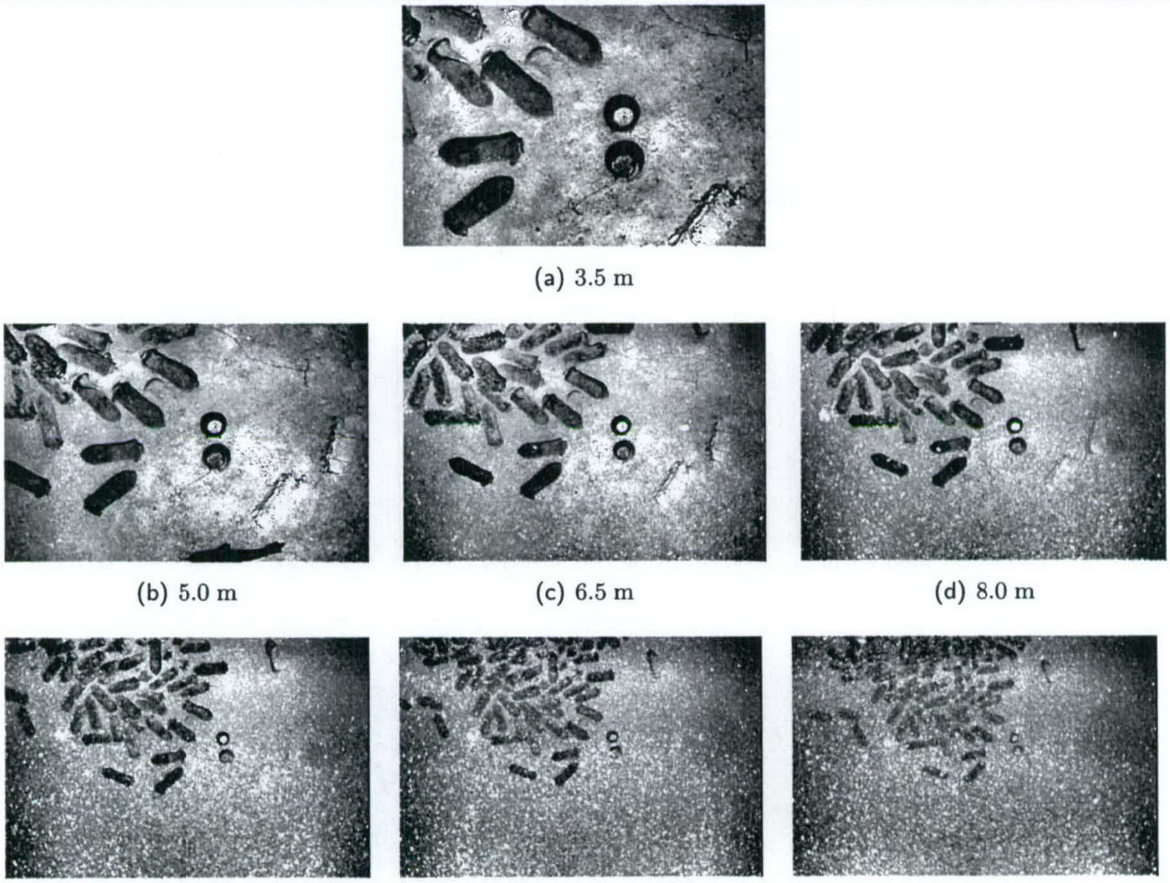


Figure 1-12 A demonstration of underwater backscatter using data collected by the Jason ROV [149]. In this example, a sequence of images is shown over an incremental range of altitudes to demonstrate the significance of backscatter in the underwater imaging process. Note that backscatter reduces the effective altitude at which an underwater vehicle can clearly image the scene. Practical camera-to-light separations for a typical AUV platform dictate that it must fly within several meters of the seafloor in order to find a tolerable tradeoff between backscatter and imaged FOV.



(a) 3.5 m

(b) 5.0 m

(c) 6.5 m

(d) 8.0 m

(e) 9.5 m

(f) 11.0 m

(g) 12.5 m

matrix or the Fundamental matrix and its goal is to recover both camera motion and scene structure [163]. In recent years, a fundamental research task has been the discovery of efficient and robust online SFM algorithms that are scalable. Early online methods approached the problem in a temporally causal manner similar to early mosaicking, and hence suffered from motion and scene drift over long sequences [8,104,181]. Their associated drift resulted from the simple open-loop construction of motion and scene which failed to use information about revisiting previously identified structure. Recently, though, researchers such as Davison [27] and Bosse [10] have begun to frame online SFM in a SLAM context, thereby benefiting from the explicit representation of joint motion/structure estimation. In particular, Davison has recently shown impressive real-time SFM results for a wearable video camera [28], though on a small spatial scale.

1.2.4 Imaging Constraints of the Underwater Environment

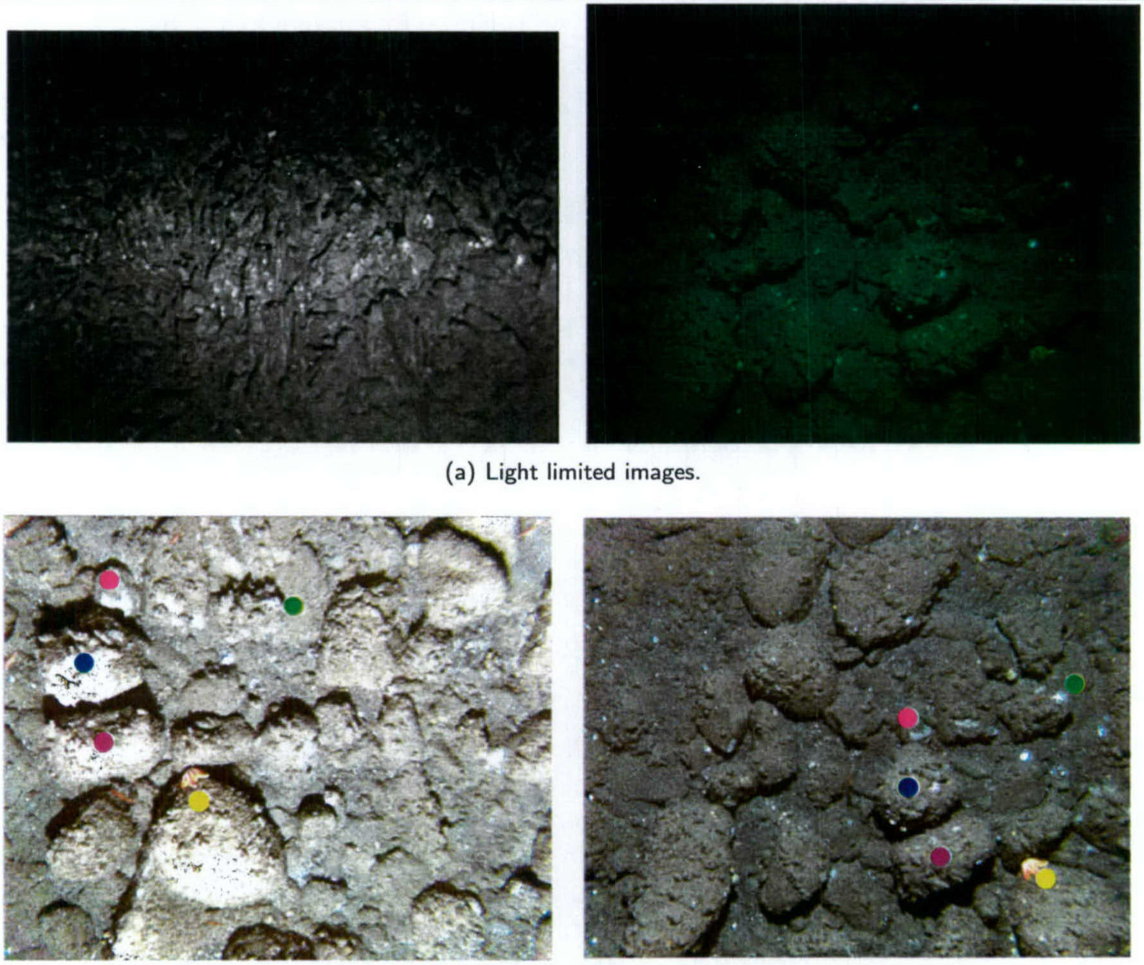
Scattering

The underwater environment places unique constraints on the ability to utilize visual information. The absorption and scattering of light through the medium of water was first understood in a physics-based context with the pioneering work by Duntley [34]. Duntley showed that the propagation of light underwater suffers from a wavelength-dependent exponential attenuation. In more recent years, McGlamery [102] investigated the fundamentals of the image formation process by computer modeling the absorption coupled with the direct, forward, and backscatter light components. Jaffe [74] later extended McGlamery's work to determine the idealized vehicle lighting configuration for minimal backscatter and good scene illumination. His results for standard lighting configurations confirmed that large horizontal camera-to-light separations were desirable to reduce backscatter — the principle cause being the reduction of common volume between the camera FOV and volume of projected light. However, Singh [149] recently showed that there are theoretical limits to the benefits of large camera-to-light separation as applied to practical vehicle configurations. Fig. 1-12 demonstrates the range over which backscatter has an effect for a fixed camera and light geometry.

Attenuation

In conjunction with the constraint of minimizing backscatter, the rapid attenuation of light through water imposes additional challenges when collecting underwater imagery (Fig. 1-13). Light attenuation limits the altitude at which a vehicle can fly from the seafloor and collect imagery. As deep-sea vehicles are required to carry their own light sources, this constraint has implications in both minimizing terrain parallax effects and in generating large-area imagery since the constraining altitude is typically 3–10 m [149]. In addition, the light source moves with the vehicle, leading to nonuniform illumination and moving shadows — both of which pose additional challenges during image registration. Vehicles are forced to fly close to the seafloor where terrain relief may be comparable to the camera to seafloor separation, inducing gross perspective changes (Fig. 1-14). Also, each image encompasses a small area of the seafloor, reducing the overall FOV. For mosaicking, this implies that many images must be registered to increase the effective FOV, and that terrain distortion

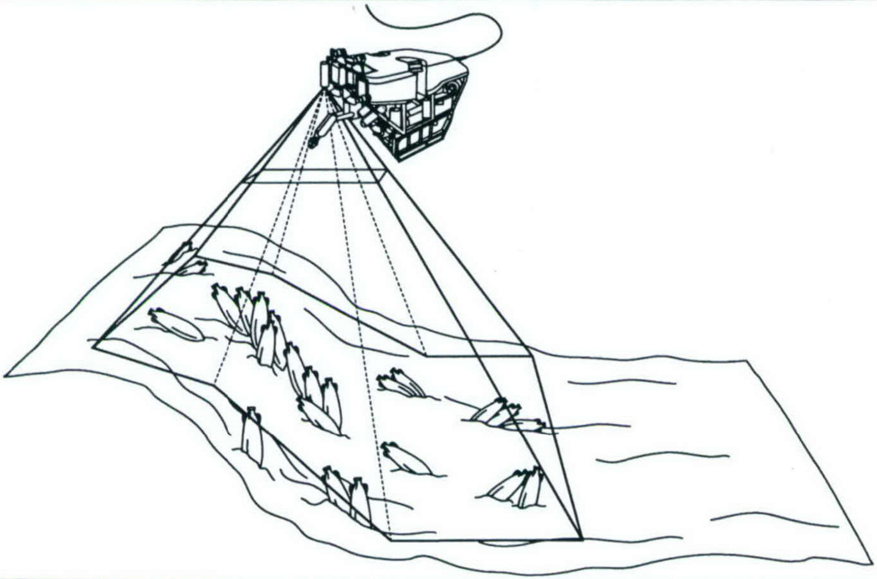
Figure 1-13 The consequences of AUVs having to carry their own light source. (a) Strobed imagery from energy constrained AUVs often tends to be light limited causing the images to appear dark and decrease in contrast towards the edges as demonstrated by the ABE imagery on the left. In addition, the preferential absorption of red light causes color images to appear green as shown by the SeaBED imagery on the right. (b) Illumination from different points can have a pronounced effect on scene appearance. This cross-track image pair was collected by the SeaBED AUV off of Stellwagen Bank National Marine Sanctuary and illustrates the effect of illumination from reciprocal headings. To aid in interpretation the rightmost image has been pre-rotated 180°to offset the nominal heading difference. Also, to improve visual queues both images have been color-corrected as described by [20] and manual correspondences have been overlaid. As an aside, note that the image on the right corresponds to the raw uncorrected color image shown above it in (a).



(a) Light limited images.

(b) Scene appearance varies markedly with location of light source.

Figure 1-14 Backscatter and light limitations dictate that the vehicle must fly close to the seafloor when collecting imagery [149]. This reduces the effective FOV for each image, requiring many images to cover a given area. Low-altitude/close flying also makes 3D terrain effects pronounced.



effects become more significant.

Registration

Image registration can also be more difficult with underwater imagery than with land-based acquired imagery. Unstructured surveys by vehicles with low-resolution navigation and heading inaccuracies are common. This results in imagery with gross motions between temporal frames, often with minimum overlap due to strobed lighting [12]. In addition, the types of imaged scenery can be vastly different ranging from highly 3D coral reefs [147] to featureless muddy bottoms [148]. Man-made features such as edges, corners, and parallel lines, prominent in land-based images, cannot be reliably expected to occur in underwater imagery.

Power

Power budget limitations of AUVs are also an important consideration in the design of imaging systems. The amount of energy expended in illuminating the scene has a direct negative effect on the endurance of these battery-limited vehicles [12]. Typically, AUVs cannot afford to put out the continuous lighting needed for video frame rates because it would come at the sacrifice of precious bottom-time. Rather, strobed lighting is often used to conserve power [147, 150]. Additionally, the low amount of image overlap afforded by this illumination scheme precludes optical-flow image registration methods such as [114,116]. Hence, the unique energy constraints of AUVs are a major driver for the goal of this thesis to be able to handle low overlap imagery (i.e., 15–35% temporal overlap).

1.3 A Review of Underwater Vehicle Navigation

The land-based community is able to obtain meter-level position accuracy almost anywhere in the world above ground via the global positioning system (GPS). However, the attenuation of electromagnetic waves through the medium of water limits the application of GPS to near surface activities. This section reviews the available and relevant techniques of underwater vehicle navigation to establish the current state-of-the-art.

1.3.1 Long Baseline Navigation

The standard for bounded XYZ navigational position measurements for underwater vehicles is the long-baseline (LBL) acoustic transponder system. LBL was originally developed at the Woods Hole Oceanographic Institution (WHOI) in the early 1970's and has since then become an integral part of the marine science community [71]. LBL requires two or more acoustic transponder beacons to be tethered to the seafloor and operates on the principle of time-of-flight. Given knowledge of the speed of sound in water (which is of the order of 1500 m/s), the round-trip travel time of an acoustic signal propagating between two unobstructed points becomes a proxy for the line of sight distance between them.

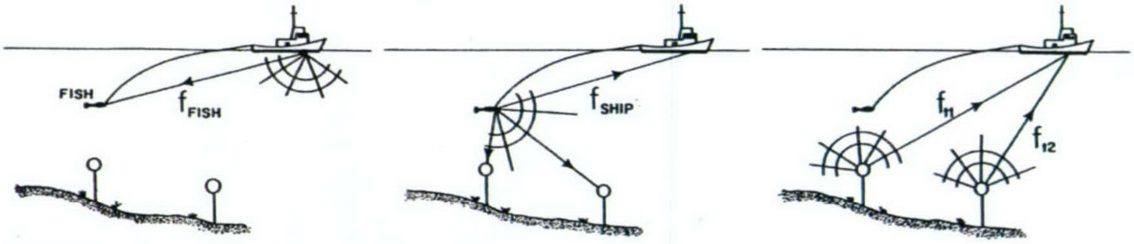
LBL Setup

The first task in setting up a LBL network is to deploy the acoustic beacons at the site of interest. This requires deploying the bottom-tethered acoustic transponders in a configuration that optimizes both the acoustic signal propagation and the geometry of the vehicle work site [71]. Next a sound velocity profile measurement is collected from the surface ship. This profile measures the sound speed throughout the water column and is used in all time-of-flight calculations to compensate for vehicle depth. Finally, each transponder in the network is surveyed and placed in a world frame of reference by the surface ship. This involves the surface ship acoustically interrogating the transponders and recording time-of-flight measurements to individual transponders while concurrently recording GPS position measurements as it steers a survey pattern from the surface. Both the recorded ship positions and transponder round-trip travel times are then processed to compute a least-squares world-referenced XYZ position for each transponder [169].

LBL Navigation and its Characteristics

A typical LBL configuration for AUVs is to have the AUV act as the master with the transponders set as slaves. The master transmits on one frequency, say 10 kHz, and upon receiving this signal each slave replies on a unique frequency, say 11 kHz and 12 kHz. Similar to the ROV cycle shown in Fig. 1-15, this allows the master to record the two-way travel time between it and each slave. A typical LBL system will operate up to a range of 10 km providing a bounded error XYZ position estimate with a range-dependent error measurement of the order of 0.1–10 m [171]. Higher frequency LBL systems operating at 300 kHz exist and are capable of higher precision position bounds [177]. Although these systems are capable of sub-centimeter position resolution, they have a maximum working range of only ~100 m as compared to the typical 12 kHz LBL systems which have a working range of

Figure 1-15 Typical LBL cycle for a ROV deployment (figure adapted from WHOI-74-6 [71]).



(a) Ship interrogates ROV at 9 kHz. (b) ROV receives interrogation at 9 kHz and replies at 10 kHz. (c) The tethered transponders receive interrogation at 10 kHz and reply at 11 and 12 kHz.

5–10 km [172]. Milne [107] provides a more detailed discussion of the implementation and working principles behind LBL and other acoustic positioning systems.

1.3.2 Doppler Velocity Logs

Recent advances have been made in the area of dead-reckoned (DR) vehicle navigation with the advent of the bottom-lock Doppler velocity log (DVL). The DVL provides a measurement of seafloor-referenced vehicle velocity, which can be integrated over time to provide XYZ positional information. The basic working principle behind these bottom-referenced velocity measurements is the acoustic Doppler effect, which states that a change in the observed sound pitch results from relative motion. This change in sound pitch is directly proportional to the relative radial velocity between the source and receiver and can be used to recover seafloor-referenced vehicle velocity. Additionally, a DVL can also be used to measure water-referenced velocities.

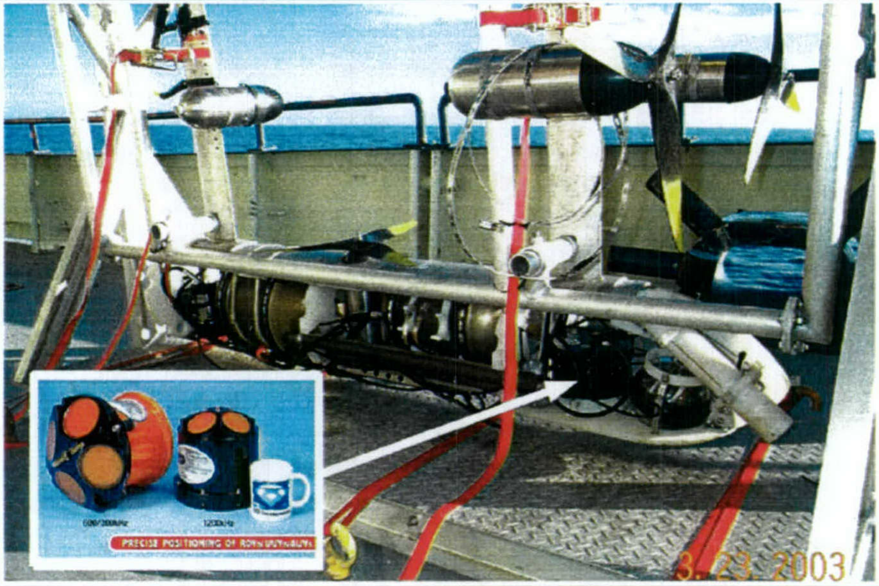
DVL Technology

Commercially available broadband DVLs (Fig. 1-16), as opposed to traditional continuous-tone DVLs, make use of time dilation to compute a velocity measurement from an ensemble of "discrete" pings. The use of time dilation results in a more accurate measurement of the Doppler shift with single ping velocity error standard deviations less than 1% [170]. When n-ping ensemble averaging is performed, the standard deviation further decreases as $\frac{1}{\sqrt{n}}$ [130]. Most off-the-shelf DVLs use a Janus transducer configuration [16], which consists of four downward-looking acoustic transducers each oriented at 30° from the vertical [130] (see inset of Fig. 1-16). In this configuration, each transducer measures the sensor's velocity with respect to the seafloor as projected onto the centerline of its acoustic beam axis, resulting in four measurements of beam-component velocity:

$$\mathbf{v}_{\mathsf{beam}}(t) = \left[v_{b_1}(t),\ v_{b_2}(t),\ v_{b_3}(t),\ v_{b_4}(t)\right]^{\top}.$$

Here, each $v_{b_i}(t)$ represents a scalar measurement of sensor velocity projected along the i^{th} beam axis (i.e., $v_{b_i}(t) = \hat{\mathbf{e}}_{b_i} \cdot \mathbf{v}_{\mathsf{sensor}}(t)$ where $\hat{\mathbf{e}}_i$ is the unit vector in the i^{th} beam direction).

Figure 1-16 A RD Instruments 1200 kHz Workhorse Navigator DVL is shown in-situ on the bottom hull of the SeaBED AUV (outer hydrodynamic shell removed).



DVL Navigation and its Characteristics

The beam component velocity measurements can be mapped to a standard Cartesian fixed instrument frame by the static 4 × 4 instrument transformation matrix M parameterized by the transducer geometry [131]:

$$\mathbf{v}_{\mathsf{sensor}}(t) = \begin{bmatrix} v_{s_x}(t) \\ v_{s_y}(t) \\ v_{s_z}(t) \\ e(t) \end{bmatrix} = \mathrm{M}\mathbf{v}_{\mathsf{beam}}(t).$$

The XYZ components of $\mathbf{v}_{\mathsf{sensor}}$ correspond to the Cartesian components of the bottom referenced velocity vector as expressed in the instrument reference frame, while $e(t)$ is a normalized least-squares measure of velocity error. Discarding the error term $e(t)$, the resulting 3-vector of instrument frame velocities, $\mathbf{v}'_{\mathsf{sensor}}(t)$, can be rotated into a locally-level coordinate frame aligned with the world frame:

$$\mathbf{v}_{\mathsf{world}}(t) = {}^{w}_{s}\mathrm{R}\mathbf{v}'_{\mathsf{sensor}}(t),$$

via the 3 × 3 rotation matrix ${}^{w}_{s}\mathrm{R}$, which is computed using measurements from onboard roll, pitch, and heading sensors. These navigation frame velocities can then be integrated to obtain a dead-reckoned bottom track DVL position [170]:

$$\mathbf{x}_{\mathsf{world}}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{v}_{\mathsf{world}}(\tau)d\tau.$$

While the dead-reckoning integration can be performed internal to the DVL using its onboard tilt and magnetic flux gate compass for orientation, it typically is computed in conjunction with the vehicle's orientation sensors for better precision. For these setups, the error dependence in the integrated vehicle position can be less than 1% of total distance traveled [16]. In [170], Whitcomb et al. define the dominant error source in DVL navigation to be heading sensor inaccuracy. They show that when augmented with absolute position 12 kHz LBL (which has error bounds on the order of 0.1–10 m) via complementary linear filtering — bounded error estimates approaching centimeter accuracy can be achieved [169].

1.3.3 Visually-Based Methods

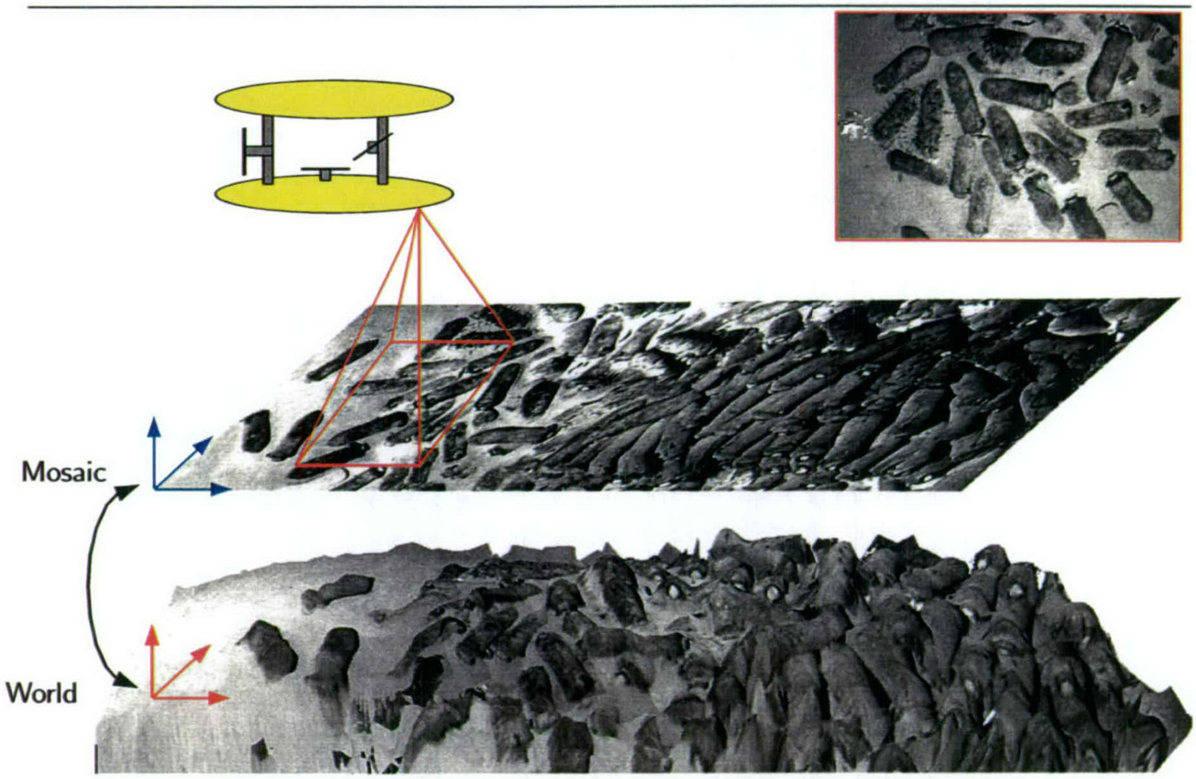
Underwater vehicles are commonly outfitted with vision sensors for biological [133, 145], geological [66,153,174,175], and archaeological [5,144,178] survey needs. As such, they have become standard equipment onboard submersibles. As a readily available sensor, vision can be incorporated into a navigation framework to provide alternative vehicle motion estimates when working near the seafloor in relatively clear water. Over the past decade, a number of techniques have been proposed within the context of real-time underwater navigation and station-keeping with one of the earliest attempts being the preliminary work of Aguirre [1] in collaboration with IFREMER (the French Institute for the Research and Exploitation of the Sea) conducted during the late 1980's. This work involved the calculation of vehicle motion from a 250 image underwater video sequence using integrated frame-to-frame translational motion estimates to provide a dead-reckoned measure of vehicle position over a small area (a few meters in size).

MBARI/Stanford

The research group at Monterey Bay Aquarium Research Institute (MBARI)/Stanford (headed by Steve Rock) further developed the role of vision as a navigation sensor [45, 46, 84, 98, 99]. For his dissertation research, Fleischer [44, 46] developed a real-time visual navigation method that exploited spatial image constraints. His strategy for reducing visual drift was to explicitly incorporate pairwise constraints from *cross-over points* (i.e., places where the vehicle's trajectory crossed back over itself) to constrain navigation error. Fleischer's algorithm was based upon a visual map representation where a collection of key image frames were stored and used to represent places the vehicle had previously visited (similar in concept to the pose-constraint network formulation of SLAM by Lu-Milios [94]). Upon revisiting one of these areas, image registration between the current view and any of the past views was used to provide 2D translational constraints between the associated camera poses. For an estimation framework, he proposed the use of either an augmented state Kalman filter or a standard linear least-squares batch formulation (in practice the batch formulation was used). Though Fleischer did not explicitly relate his work to the SLAM literature, it was definitely in the same vein. Limitations of the approach, however, are the overly simplistic 2D translation-only image registration model and the non-scalable estimation framework.[1]

[1] To satisfy the assumptions of image-based correlation, the vehicle was actively controlled to maintain a fixed heading (using compass readings) and flew only over flat portions of seafloor at a constant altitude.

Figure 1-17 A mosaic-based navigation strategy. The vehicle navigates with respect to the mosaic by registering its current view to the mosaic, thereby achieving a bounded error pose estimate. Pitfalls of this method are: 1) mosaic construction is ill-posed over non-planar terrain and 2) it is an awkward mathematical framework for fusing other sensor-based navigation measurements.



University of Miami

Simultaneously, Negahdaripour et al. [114–116] at the University of Miami, Florida developed a mosaic-based navigation (MBN) strategy for video imagery using an optical flow method founded upon the generalized BCC [113, 117] (similar to the BCC constraint of §1.2.1, but additionally accounting for affine varying illumination). Fig. 1-17 illustrates the concept behind the MBN strategy. Their differential formulation uses spatio-temporal image gradients to measure inter-frame vehicle motion directly, then integrates over time to provide an estimate of vehicle position. To reduce the drift rate of the navigation estimate, the authors offer two methods. The first is based upon trying to calculate the biases associated with their direct method in an attempt to improve the inter-image motion estimate and thereby reduce drift — this strategy is analogous to using a better inertial navigation unit (INU) (i.e., the associated measurement errors are reduced, but not eliminated). Hence, it does not avoid the undesired characteristic of unbounded error growth. Their second modification attempts to address unbounded error growth by using the mosaic itself to help constrain position drift. Correction of errors in position and orientation are made each time the mosaic is updated, which occurs every L^{th} video frame. They use their current position estimate $\hat{P}[k]$ to extract the mosaic region $M[k]$ where the current image

$I[k+1]$ is hypothesized to map to, producing the estimated image $\hat{I}[k+1]$. The motion error estimate computed from $I[k+1]$ and $\hat{I}[k+1]$ is then used as a feedback to correct the current position $\hat{P}[k+1]$ and update the mosaic [116]. With this method, error growth is now constrained to the accuracy of the mosaic.

Instituto Superior Técnico

Finally, the work of Gracias et al. [56–58] at the Instituto Superior Técnico, Portugal also approaches the problem of local AUV navigation in a mosaic-based manner. In their MBN approach, the video mosaic is first generated *offline*, then used online for real-time navigation. The offline mosaic is constructed in a globally consistent manner taking into account spatial pairwise constraints, and under the assumption that the imaged terrain is approximately planar, their method uses the mosaic map for navigation. They do this by decomposing image-to-mosaic homographies into camera poses using the world plane decomposition of Faugeras and Lustman [41]. While their approach has been successfully demonstrated for small areas (i.e., a navigable area covering approximately 64 m^2 with a 10.8 m×9.5 m bounding box [58]), it does not scale well to large areas because it suffers from inconsistencies associated with generating a single, large, internally consistent mosaic map that meets the assumption of an extended planar scene. In addition, another drawback of their method is that it is a purely vision based framework awkward for fusing other sensor based navigation measurements.

1.3.4 Limitations of Current Approaches

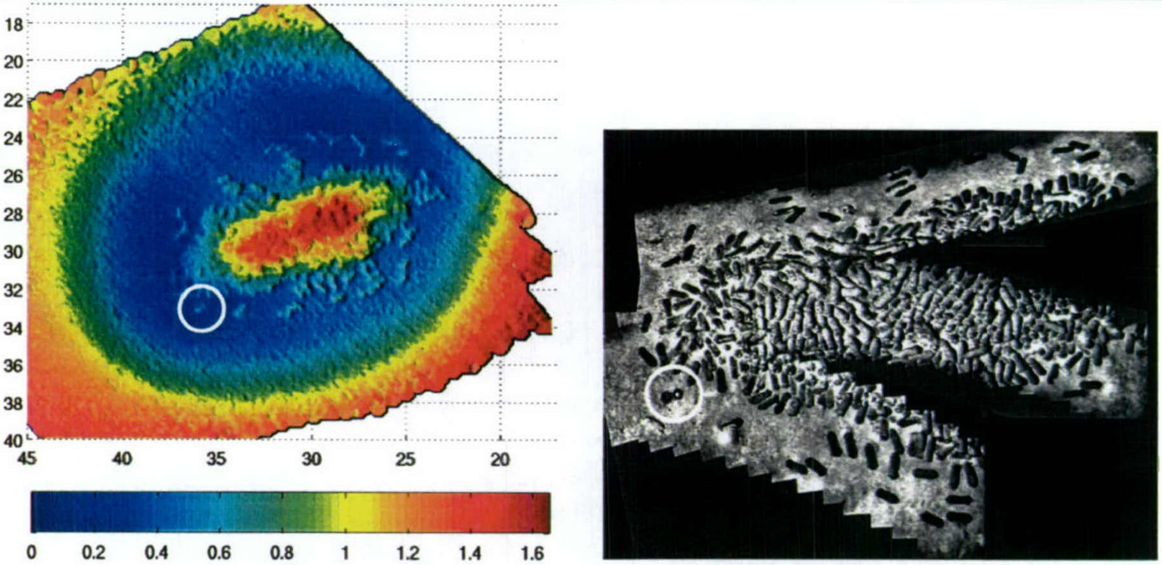
Acoustic transponder navigation systems offer bounded position measurements, but at high deployment costs — both in the context of ship deployment time and equipment costs. Additionally, such systems limit vehicle navigation to within the deployed network, which is not amenable to conducting multiple short-duration exploratory surveys over different sites. Alternatively, the recent advent of the DVL reduces the need for additional infrastructure by improving the dead-reckon navigation capability of near-seafloor vehicles. However, the open-loop nature of these systems implies that error is unbounded as a function of distance traveled.

Turning to more recent underwater visually-based methods, the dominant approach has been the mosaic-based navigation strategy. While current MBN implementations have been demonstrated to have practical application over small, relatively flat, areas (hundreds of square meters), the following limitations make MBN unsuitable for large-area navigation:

1. Current approaches are founded solely from a computer vision standpoint (i.e., all motion estimates are derived from imagery without incorporating additional motion information from available vehicle navigation sensors). While this is interesting from a theoretical basis, the problem begs to be formulated within an estimation framework that fuses all available vehicle information to produce an optimal solution.

2. Construction of a single large mosaic is ill-conditioned when the seafloor deviates from the planar assumption (Fig. 1-18). A piecewise planar submapping strategy could possibly be employed in an attempt to salvage MBN. However, highly 3D

Figure 1-18 Failure of a MBN strategy lies in the ill-posed construction of a single large mosaic over non-planar terrain. Here we see bathymetry from a deep-water archaeological site showing an amphora pile sitting in a small depression and that construction of a mosaic over the same pile fails due to 3D relief [149] (the circle designates the same area in both modalities).



(a) Bathymetry map over a deep-water amphora pile (units are in meters).

(b) The mosaic diverges as a result of 3D terrain effects violating the planar seafloor assumption.

environments (such as surveys over coral reefs [145] where mosaicking assumptions are severely violated) would remain inapplicable.

3. Low overlap imagery is common for AUVs since they are power limited and cannot afford the continuous illumination necessary for video frame rates [12, 152]. Hence, image registration is much more difficult in this scenario since the inter-image motion may be large. This implies that direct methods and video-rate techniques are inapplicable because they assume incremental (i.e., high overlap) camera motion between frames.

These limitations, in conjunction with the limitations of traditional LBL and DVL methods, have been the impetus for the integrated systems-level visual approach adopted in this thesis.

1.4 Thesis Outline

In this section we outline our systems-level approach to visually-based navigation and the order in which material is presented in this thesis document.

1.4.1 Approach

This thesis answers the question of how to achieve real-time, scalable, bounded error, 6-DOF, precision navigation for near-seafloor AUVs in rugged-terrain via the incorporation of camera-derived motion estimates. Unlike prevailing MBN methods described above, this thesis adopts a systems-level approach to using visual navigation underwater, termed "visually augmented navigation (VAN)" [36]. Specifically, VAN casts visual sensor fusion within a stochastic view-based SLAM framework amenable to the peculiarities of low-overlap underwater imagery. VAN combines pairwise image-derived relative-pose spatial constraints with more traditional navigation sensor measurements (e.g., attitude, Doppler velocities, depth) to recover a bounded estimate of the vehicle's 6-DOF trajectory. In this context, VAN embraces the stochastic framework of SLAM while addressing the practical issue of "features" and map representation in an unstructured underwater environment observed through low-overlap imagery. Rather than detecting and tracking particular features over time, overlapping *image pairs* are registered in a wide-baseline epipolar framework to recover the relative orientation and baseline direction between camera poses. These pairwise image-derived spatial constraints effectively allow virtual observation of the current vehicle pose, $\mathbf{x}_v$, relative to any other pose, $\mathbf{x}_j$, with common scene overlap resulting in a view-based map of the world where the observation model is of the form $\mathbf{h}(\mathbf{x}_v, \mathbf{x}_j)$ — thus, vehicle poses effectively become virtual "landmarks" [36, 87].

1.4.2 Document Roadmap

The material presentation is broken into the following chapters.

- **Chapter 2** lays the foundation of an estimation framework suitable for fusing low-overlap pairwise image-derived constraints. Additionally, we present a systems-level image registration strategy that exploits the available information from a proprioceptive/exteroceptive pose-instrumented robotic platform.

- **Chapter 3** discusses the "information formulation" of the feature-based SLAM posterior. The recent popularity of this representation stems from its "almost sparse" structure. In particular, a number of recent large-area SLAM algorithms have been derived by enforcing sparsity in this representation. We explore the consequences of this approximation and provide new insight as to why, and how, this can lead to filter inconsistency.

- **Chapter 4** presents the novel insight that the information formulation can be made exactly sparse without any approximation by using a view-based SLAM representation. This allows us to scale the estimation framework of Chapter 2 to very large environments achieving $\mathcal{O}(n)$ complexity by exploiting the sparse representation. Additionally, we present a novel algorithm for data association — something that had previously been an open research issue in the information form.

- **Chapter 5** provides a summary of contributions, algorithm failure modes, and suggestions for future work.

- **Appendix A** describes in detail our robot platform and the various analytical models used to describe it.

- **Appendix B** provides accompanying derivations for the work presented in Chapter 3.

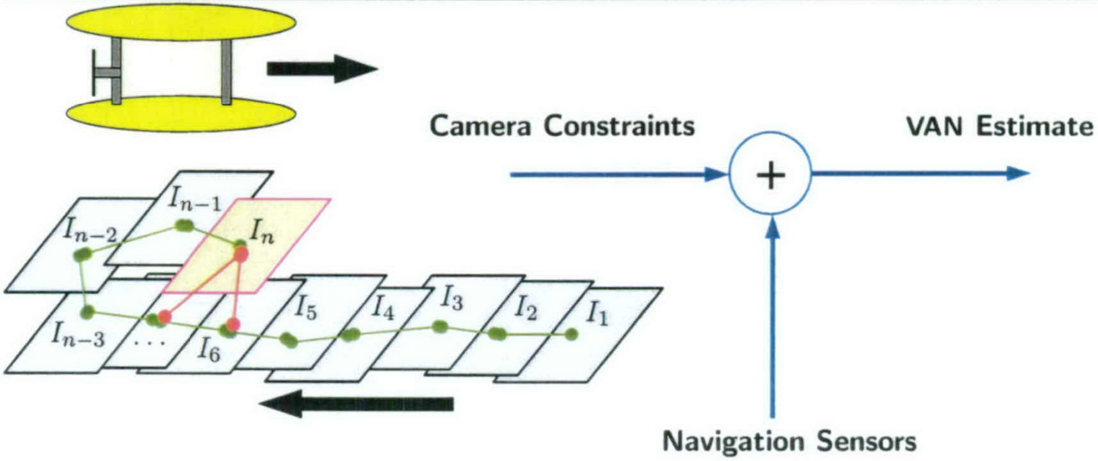
CHAPTER 2 Visually Augmented Navigation

As autonomous underwater vehicles are used more routinely in an exploratory context for ocean science, the goal of visually augmented navigation is to improve the near-seafloor navigation precision of such vehicles without imposing the burden of having to deploy additional infrastructure. This is in contrast to traditional acoustic long baseline navigation techniques, which require the deployment, calibration, and eventual recovery of a transponder network. To achieve this goal, VAN is framed within a vision-based simultaneous localization and mapping framework that exploits the systems-level complementary aspects of a camera and strap-down sensor suite to create a bounded-error navigation technique that is robust to the peculiarities of low-overlap underwater imagery. It uses a view-based representation where camera-derived relative-pose measurements provide spatial constraints, which enforce trajectory consistency and also serve as a mechanism for loop-closure (Fig. 2-1). This chapter outlines the multi-sensor VAN framework and demonstrates it to have compelling advantages over a purely vision-only based approach by 1) improving the robustness of low-overlap underwater image registration, 2) setting the free gauge scale, and 3) allowing for a disconnected camera constraint topology.

2.1 Introduction

From exploring abandoned mines in Pennsylvania [159], to exploring other planets in our solar-system [26], robotic exploration of remote environments extends our reach to areas where human exploration is considered too dangerous, too technically challenging, or both. While high profile missions like the 2003 Mars rovers epitomize the lengths that we will go to in search of new origins of life, it cannot be overstated that exploring the deep-abyss of our own oceans can be nearly as alien and offer just as startling discoveries about how life began. Though manned vehicles like Alvin [2, 31] have been responsible for many of the most important deep-science discoveries [4, 25], the extreme design requirements, operational costs, risk of life, and limited availability prevent its ubiquitous use. Therefore, out of necessity the deep-sea has become an arena where the presence of mobile robotics is

Figure 2-1 The objective of VAN is the real-time fusion of "zero-drift" camera measurements with navigation sensor data to close-the-loop on dead-reckoned error. For this purpose VAN adopts a top-down systems-level approach to visual navigation. It uses a view-based representation founded upon registering raw imagery to generate pairwise camera constraints that are then fused with navigation sensor data in a delayed-state extended Kalman filter framework.



pervasive and their scientific utility revolutionary [6, 141, 145, 176].

While underwater mobile robotics have made significant inroads into mainstream science over the past two decades, a limiting technological issue to their widespread utility, especially for exploration, is the lack of *easily* obtainable precision navigation. With the advent of GPS many surface and air vehicle applications are able to easily obtain their position anywhere on the globe with precision of a few meters via the triangulation of satellite transmitted radio signals — unfortunately, these radio signals do not penetrate sub-sea [172] (nor underground [159], or even indoors [19]). Therefore, as discussed in Chapter 1, traditional underwater navigation strategies have been to deploy an acoustic version of GPS using seafloor tethered beacons to relay time-of-flight range measurements for triangulated positioning. However, the cost, complexity, and limitations of an infrastructure dependent solution leave much to be desired, which is further complicated by the fact that alternative strap-down solutions suffer from a position drift that grows unbounded with time.

Over the past decade, a big research push within the terrestrial mobile robotics community has been to develop environmentally-based navigation algorithms, which eliminate the need for additional infrastructure and bound position error growth to the size of the environment — a key prerequisite for truly autonomous navigation. The basis of this work has been to exploit the perceptual sensing capabilities of robots to "beat-down" accumulated odometric error by localizing the robot with respect to "landmarks" in the environment. The question of how to use such a methodology for navigation and mapping was first theoretically addressed in a probabilistic framework in the mid 1980's with seminal papers by Smith, Self, and Cheeseman [154] and Moutarlier and Chatila [110], which have since then become the cornerstone of the research field known as SLAM.

One of the major challenges of a SLAM methodology is that defining what constitutes a feature from raw sensor data can be nontrivial. In man-made structured environments, typically composed of planes, lines and corners primitives, features can be more easily defined [158]. However, unstructured outdoor environments can pose a more challenging task for feature extraction and matching, which has lead to "scan-matching" [94] based approaches that do not require an explicit representation of features. These view-based techniques have traditionally been used with accurate perceptual sensors such as laser range finders where raw data can be "matched" directly in an iterative closest point sense. Along these lines, our underwater approach is to use a camera as an accurate and inexpensive perceptual sensor to collect near-seafloor imagery that can also be "matched" directly. Motivation for such an approach comes from the fact that typical AUV imagery has low temporal overlap (to minimize illumination power consumption [12]), which implies that 3D features in the environment are not observed for more than a few views. Such a low-overlap constraint implies that a view-based representation is particularly suitable for this type of data since overlapping image pairs can be registered directly in a pairwise fashion to extract "zero-drift" relative-pose modulo scale measurements without explicitly representing 3D feature points. In this way, registering an image taken from time t_i to an image taken at time t_j provides a spatial constraint whose error is bounded regardless of time or distance traveled between the two views.

In the rest of this chapter we present our framework and methodology for incorporating camera-derived relative-pose measurements with vehicle navigation data in a view-based SLAM context. In particular, §2.2 describes our assumptions while §2.3 presents a delayed-state extended Kalman filter (EKF) framework for fusing camera measurements that also serves as a foundation for probabilistic link hypothesis. In §2.4 we then explain how we actually make the pairwise camera measurements using a systems-level feature-based image registration approach. We show that a multi-sensor approach has compelling advantages over a camera-only based navigation system and in particular that it improves registration robustness via a pose-constrained correspondence search. Results are presented in the context of a real-world dataset collected by an AUV in a rugged undersea environment, and for tank data collected by a ROV for which ground-truth was available.

2.2 Assumptions

Our application is based upon using a pose instrumented AUV equipped with a single down-looking calibrated camera to perform underwater imaging and mapping [146, 147]. The vehicle makes acoustic measurements of both velocity and altitude relative to the seafloor. Absolute orientation is measured to within a few degrees over the entire survey area via inclinometers and a flux-gate magnetic compass. Bounded positional estimates of depth, z, are provided by a pressure sensor. A detailed platform discussion can be found in §A.1, however, for convenience Table 2.1 provides a short summary of assumed sensor characteristics. In brief we assume:

- An ideal, calibrated camera.
- An instrumented platform.
- Known reference frames (i.e., vehicle to camera, vehicle to sensor).
- Pairwise registration to accommodate low-temporal overlap.

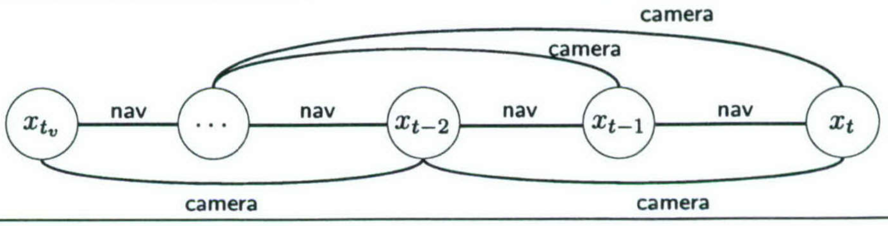
Table 2.1 Typical pose sensor characteristics for underwater platforms.

Measurement	Sensor	Precision
Roll/Pitch	Tilt Sensor	±0.5°
Heading	Magnetic Flux Gate Compass	±2.0°
3-Axis Angular Rate	AHRS	±1.0°/s
Body Frame Velocities	Acoustic Doppler	±1–2 mm/s
Depth	Pressure Sensor	±0.01%
Altitude	Acoustic Altimeter	±0.1 m

2.3 View-Based SLAM Estimation Framework

Typical structure-from-motion approaches [8,27,104,163,181] estimate both camera motion and 3D scene structure from a sequence of video frames. However, in our application the low degree of temporal image overlap (typically on the order of 35% or less) motivates us to focus on recovering pairwise measurements from spatially neighboring image frames. In this framework, the camera provides measurements of the 6-DOF relative coordinate transformation between poses modulo scale. These measurements are used as constraints in a recursive estimation framework that tries to determine the global poses consistent with the camera measurements and navigation prior as shown in Fig. 2-2. These global poses correspond to samples from the robot's trajectory at the times associated with image acquisition. Thus, unlike the typical feature-based SLAM estimation problem, which keeps track of the current robot pose and an associated landmark map, the VAN state vector consists entirely of delayed vehicle states corresponding to the vehicle poses at the times the images were captured. This delayed-state approach corresponds to a view-based representation of the environment, which can be traced back to a batch scan-matching method by Lu and Milios [94] using laser data, a delayed decision making framework by Leonard and Rikoski [87] for feature initialization with sonar data, and the hybrid batch/recursive formulations by Fleischer [44] and McLauchlan [105] using camera images. In this context, scan-matching raw images results in virtual observations of robot motion with respect to a place it has previously visited.

Figure 2-2 A view-based representation consists of a network of navigation and camera constraints over a collection of time-delayed vehicle poses associated with the images in our view-based map.



2.3.1 Delayed-State Extended Kalman Filter

We begin by describing our representation of vehicle state and a general system model for state evolution and observation.[1] This model is used as the basis for trajectory estimation within the context of an EKF [54]. We then show how to incorporate camera-derived relative-pose measurements by augmenting our state representation to include a history of delayed-state vehicle poses.

Fixed-Size State Description

The vehicle state vector, $\mathbf{x}_v$, contains both pose and kinematic terms, $\mathbf{x}_p$ and $\mathbf{x}_\kappa$ respectively, and is defined as

$$\mathbf{x}_v \equiv \left[\mathbf{x}_p^\top, \quad \mathbf{x}_\kappa^\top\right]^\top.$$

Here $\mathbf{x}_p$ is a 6-vector of vehicle pose in the local-level navigation frame where XYZ roll pitch heading Euler angles are used to represent orientation [48] (i.e., $\mathbf{x}_p \equiv \mathbf{x}_{\ell v} \equiv \left[x, y, z, \phi, \theta, \psi\right]^\top$), and $\mathbf{x}_\kappa$ represents any kinematic state elements that are required for propagation of the vehicle process model (e.g., body-frame velocities, accelerations, angular rates). In addition we assume that the vehicle state can be modeled as being normally distributed, $\mathbf{x}_v \sim \mathcal{N}\left(\boldsymbol{\mu}_v, \Sigma_{vv}\right)$, with mean and covariance given by

$$\boldsymbol{\mu}_v = \left[\boldsymbol{\mu}_p^\top, \quad \boldsymbol{\mu}_\kappa^\top\right]^\top \quad \text{and} \quad \Sigma_{vv} = \begin{bmatrix} \Sigma_{pp} & \Sigma_{p\kappa} \\ \Sigma_{\kappa p} & \Sigma_{\kappa\kappa} \end{bmatrix}.$$

The vehicle state evolves through a time-varying continuous time process model, $\mathbf{f}(\,\cdot\,, t)$, driven by white noise, $\mathbf{w}(t) \sim \mathcal{N}\left(\mathbf{0}, \mathrm{Q}(t)\right)$, and deterministic control inputs, $\mathbf{u}(t)$, while discrete time measurements of elements in the vehicle state are observed through an observation model, $\mathbf{h}(\,\cdot\,, t_k)$, corrupted by time independent Gaussian noise, $\mathbf{v}[t_k] \sim \mathcal{N}\left(\mathbf{0}, \mathrm{R}_k\right)$, with $E\left[\mathbf{w}\mathbf{v}^\top\right] = 0$. The resulting system model is:

$$\begin{aligned} \dot{\mathbf{x}}_v(t) &= \mathbf{f}(\mathbf{x}_v(t), \mathbf{u}(t), t) + \mathbf{w}(t) \\ \mathbf{z}[t_k] &= \mathbf{h}(\mathbf{x}_v[t_k], t_k) + \mathbf{v}[t_k]. \end{aligned} \tag{2.1}$$

As is typical in the navigation literature, the vehicle state distribution is approximately maintained using a continuous-discrete EKF [54] given by

$$\boxed{\textbf{Prediction}} \qquad \begin{aligned} \dot{\boldsymbol{\mu}}_v(t) &= \mathbf{f}(\boldsymbol{\mu}_v(t), \mathbf{u}(t), t) \\ \dot{\Sigma}_{vv}(t) &= \mathrm{F}_\mathrm{x}\Sigma_{vv}(t) + \Sigma_{vv}(t)\mathrm{F}_\mathrm{x}^\top + \mathrm{Q}(t) \end{aligned} \tag{2.2}$$

$$\boxed{\textbf{Update}} \qquad \begin{aligned} \mathrm{K} &= \bar{\Sigma}_{vv}\mathrm{H}_\mathrm{x}^\top\left(\mathrm{H}_\mathrm{x}\bar{\Sigma}_{vv}\mathrm{H}_\mathrm{x}^\top + \mathrm{R}_k\right)^{-1} \\ \boldsymbol{\mu}_v &= \bar{\boldsymbol{\mu}}_v + \mathrm{K}\left(\mathbf{z}[t_k] - \mathbf{h}(\bar{\boldsymbol{\mu}}_v, t_k)\right) \\ \Sigma_{vv} &= \left(\mathrm{I} - \mathrm{K}\mathrm{H}_\mathrm{x}\right)\bar{\Sigma}_{vv}\left(\mathrm{I} - \mathrm{K}\mathrm{H}_\mathrm{x}\right)^\top + \mathrm{K}\mathrm{R}_k\mathrm{K}^\top \end{aligned} \tag{2.3}$$

where $\mathrm{F}_\mathrm{x} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_v}\Big|_{\boldsymbol{\mu}_v}$ and $\mathrm{H}_\mathrm{x} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_v}\Big|_{\bar{\boldsymbol{\mu}}_v}$ are the process and observation model Jacobians re-

[1] See Appendix A for details.

spectively. In this formulation, the predicted vehicle distribution, $\bar{\mathbf{x}}_v \sim \mathcal{N}\big(\bar{\boldsymbol{\mu}}_v, \bar{\Sigma}_{vv}\big)$, is computed between asynchronous sensor measurements by solving (2.2) via a fourth-order Runge-Kutta numerical integration approach [119].

Unfortunately, the fixed-size state description, $\mathbf{x}_v$, does not allow us to represent our pairwise camera constraints. This is because registration of an image pair results in a relative-pose measurement modulo scale, and not an absolute observation of elements in vehicle pose, $\mathbf{x}_p$. Therefore, before we can incorporate pairwise camera constraints, we have to first augment our state representation to include a history of vehicle poses where each delayed-state entry corresponds to an image in our view-based map. Under this representation, the distribution we are trying to estimate is $p\big(\boldsymbol{\xi}_t \big| \mathbf{z}^t, \mathbf{u}^t\big)$ where $\mathbf{z}^t$ represents all measurements up to time t (including camera and navigation sensors), $\mathbf{u}^t$ is the set of all control inputs, and $\boldsymbol{\xi}_t$ is our view-based SLAM state vector (note that initially $\boldsymbol{\xi}_t \equiv \mathbf{x}_v$). Next, we describe the process of *how* delayed-states are added to our "map".

Augmenting our State Description with Delayed-States

At time t_1 corresponding to when the first image frame, I_1, is captured, we augment our state description, $\boldsymbol{\xi}_t$, to include the vehicle's pose of where it was when it took that image (i.e., $\boldsymbol{\xi}_t = [\mathbf{x}_v^\top, \mathbf{x}_{p_1}^\top]^\top$). Therefore, at this time instance the augmented state distribution, $\boldsymbol{\xi}_t \sim \mathcal{N}\big(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\big)$, is given by

$$\begin{aligned} \boldsymbol{\mu}_t &= \big[\boldsymbol{\mu}_v[t_1]^\top,\ \boldsymbol{\mu}_p[t_1]^\top\big]^\top &&\equiv \big[\boldsymbol{\mu}_v^\top,\ \boldsymbol{\mu}_{p_1}^\top\big]^\top \\ \Sigma_t &= \begin{bmatrix} \Sigma_{vv}[t_1] & \Sigma_{vp}[t_1] \\ \Sigma_{vp}^\top[t_1] & \Sigma_{pp}[t_1] \end{bmatrix} &&\equiv \begin{bmatrix} \Sigma_{vv} & \Sigma_{vp_1} \\ \Sigma_{p_1 v} & \Sigma_{p_1 p_1} \end{bmatrix}. \end{aligned} \tag{2.4}$$

This process is repeated for each camera frame that we wish to keep in our view-based map so that after augmenting n delayed states (one for each camera frame) we have $\boldsymbol{\xi}_t = \big[\mathbf{x}_v^\top,\ \mathbf{x}_{p_1}^\top,\ \cdots,\ \mathbf{x}_{p_n}^\top\big]^\top$ with

$$\boldsymbol{\mu}_t = \begin{bmatrix} \boldsymbol{\mu}_v \\ \boldsymbol{\mu}_{p_1} \\ \vdots \\ \boldsymbol{\mu}_{p_n} \end{bmatrix} \quad \text{and} \quad \Sigma_t = \begin{bmatrix} \Sigma_{vv} & \Sigma_{vp_1} & \cdots & \Sigma_{vp_n} \\ \Sigma_{p_1 v} & \Sigma_{p_1 p_1} & \cdots & \Sigma_{p_1 p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{p_n v} & \Sigma_{p_n p_1} & \cdots & \Sigma_{p_n p_n} \end{bmatrix}. \tag{2.5}$$

Note that in (2.4) the vehicle's current pose $\mathbf{x}_p$ is fully correlated with $\mathbf{x}_{p_1}$ by definition. Therefore, when the n^{th} delayed-state, $\mathbf{x}_{p_n}$, is augmented in (2.5), its cross-correlation with the other delayed poses in Σ_t is non-zero since the current vehicle state has correlation with each delayed-state.

The system model (2.1) must be also be extended to incorporate the augmented state representation. For the process model the only required change is that $\mathbf{x}_v$ continues to evolve through the vehicle dynamic model, $\mathbf{f}(\,\cdot\,, t)$, while the delayed-state entries do not:

$$\dot{\boldsymbol{\xi}}_t = \frac{d}{dt}\begin{bmatrix} \mathbf{x}_v \\ \mathbf{x}_{p_1} \\ \vdots \\ \mathbf{x}_{p_n} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_v(t), \mathbf{u}(t), t) + \mathbf{w}(t) \\ 0_{6\times 1} \\ \vdots \\ 0_{6\times 1} \end{bmatrix}.$$

Similarly, navigation sensor observation models continue to remain a function of only the current vehicle state, $\mathbf{x}_v$, which results in sparse Jacobians of the form

$$\mathrm{H}_{\xi} = \begin{bmatrix} \mathrm{H}_{\mathrm{x}}, & 0_{m\times 6}, & \cdots, & 0_{m\times 6} \end{bmatrix}$$

where m is the dimension of the measurement. However, in the case of camera-derived measurements the observation model becomes a function of delayed-states entries as is discussed next.

2.3.2 Pairwise Camera Observation Model

Pairwise image registration from a calibrated camera has the ability to provide a measurement of relative-pose modulo scale between delayed-state elements $\mathbf{x}_{p_i}$ and $\mathbf{x}_{p_j}$, provided images I_i and I_j have overlap. In deriving the camera observation model we use the familiar Smith, Self, and Cheeseman notation [154] described in §A.2, and assume that the camera to vehicle static pose $\mathbf{x}_{vc}$ is known.

Camera Relative Pose

The delayed-state entries $\mathbf{x}_{p_i}$ and $\mathbf{x}_{p_j}$ correspond to vehicle poses $\mathbf{x}_{\ell v_i}$ and $\mathbf{x}_{\ell v_j}$ as represented in the local-level navigation frame respectively. Hence, using the static camera to vehicle pose, $\mathbf{x}_{vc}$, we can express the transformation from camera frame i to j using the tail-to-tail operation as

$$\begin{aligned} \mathbf{x}_{c_j c_i} &= \ominus \mathbf{x}_{\ell c_j} \oplus \mathbf{x}_{\ell c_i} && (2.6a) \\ &= \ominus(\mathbf{x}_{\ell v_j} \oplus \mathbf{x}_{vc}) \oplus (\mathbf{x}_{\ell v_i} \oplus \mathbf{x}_{vc}) && (2.6b) \end{aligned}$$

with Jacobian

$$\begin{aligned} \mathrm{J}_{cjci} &= \frac{\partial \mathbf{x}_{cjci}}{\partial(\mathbf{x}_{\ell v_j}, \mathbf{x}_{\ell v_i})} = \underbrace{\frac{\partial \mathbf{x}_{cjci}}{\partial(\mathbf{x}_{\ell c_j}, \mathbf{x}_{\ell c_i})} \cdot \frac{\partial(\mathbf{x}_{\ell c_j}, \mathbf{x}_{\ell c_i})}{\partial(\mathbf{x}_{\ell v_j}, \mathbf{x}_{\ell v_i})}}_{\text{chain-rule}} && (2.7a) \\ &= {}_{\ominus}\mathrm{J}_{\oplus}\big|_{(\mathbf{x}_{\ell c_j}, \mathbf{x}_{\ell c_i})} \cdot \begin{bmatrix} \mathrm{J}_{\oplus 1}\big|_{(\mathbf{x}_{\ell v_j}, \mathbf{x}_{vc})} & 0_{6\times 6} \\ 0_{6\times 6} & \mathrm{J}_{\oplus 1}\big|_{(\mathbf{x}_{\ell v_i}, \mathbf{x}_{vc})} \end{bmatrix}. && (2.7b) \end{aligned}$$

5-DOF Camera Measurement

However, note that what the camera actually measures is not the 6-DOF relative pose measurement of (2.6), but rather only a 5-DOF measurement due to loss of scale in the image formation process. This loss of scale implies that only the baseline direction, as represented by azimuth and elevation angles α_{ji} and β_{ji} respectively, is recoverable from image space. Realizing that the relative-pose measurement $\mathbf{x}_{c_j c_i}$ is parameterized by

$$\mathbf{x}_{c_j c_i} = [{}^{c_j}\mathbf{t}_{c_j c_i}^{\top}, \mathbf{\Theta}_{c_j c_i}^{\top}]^{\top} = [x_{ji}, y_{ji}, z_{ji}, \phi_{ji}, \theta_{ji}, \psi_{ji}]^{\top},$$

we can express the bearing-only baseline measurement of α_{ji} and β_{ji} as

$$\alpha_{ji} = \operatorname{atan2}(y_{ji}, x_{ji}) \qquad\qquad \beta_{ji} = \operatorname{atan2}\big(z_{ji}, (x_{ji}^2 + y_{ji}^2)^{\frac{1}{2}}\big)$$

with Jacobian[2]

$$\begin{aligned} \mathrm{J}_{\alpha\beta} &= \frac{\partial(\alpha_{ji}, \beta_{ji})}{\partial^{c_j}\mathbf{t}_{c_j c_i}} \\ &= \begin{bmatrix} \frac{-y_{ji}}{(x_{ji}^2+y_{ji}^2)} & \frac{x_{ji}}{(x_{ji}^2+y_{ji}^2)} & 0 \\ \frac{-z_{ji}x_{ji}}{(x_{ji}^2+y_{ji}^2)^{1/2}(x_{ji}^2+y_{ji}^2+z_{ji}^2)} & \frac{-z_{ji}y_{ji}}{(x_{ji}^2+y_{ji}^2)^{1/2}(x_{ji}^2+y_{ji}^2+z_{ji}^2)} & \frac{(x_{ji}^2+y_{ji}^2)^{1/2}}{(x_{ji}^2+y_{ji}^2+z_{ji}^2)} \end{bmatrix}. \end{aligned}$$

Hence, the pairwise 5-DOF camera observation model becomes

$$\mathbf{z}_{ji} = \mathbf{h}_{ji}(\boldsymbol{\xi}_t) = \mathbf{h}_{ji}(\mathbf{x}_{p_j}, \mathbf{x}_{p_i}) = [\alpha_{ji}, \beta_{ji}, \phi_{ji}, \theta_{ji}, \psi_{ji}]^{\top} \tag{2.8}$$

with Jacobian

$$\mathrm{H}_{\xi} = \begin{bmatrix} 0 \cdots & \frac{\partial \mathbf{h_{ji}}}{\partial \mathbf{x}_j} & \cdots 0 \cdots & \frac{\partial \mathbf{h_{ji}}}{\partial \mathbf{x}_i} & \cdots 0 \end{bmatrix}$$

where

$$\frac{\partial \mathbf{h}_{ji}}{\partial(\mathbf{x}_{p_j}, \mathbf{x}_{p_i})} = \frac{\partial \mathbf{h}_{ji}}{\partial \mathbf{x}_{c_j c_i}} \cdot \frac{\partial \mathbf{x}_{c_j c_i}}{\partial(\mathbf{x}_{p_j}, \mathbf{x}_{p_i})} = \begin{bmatrix} \mathrm{J}_{\alpha\beta} & 0_{2\times3} \\ 0_{3\times3} & \mathrm{I}_{3\times3} \end{bmatrix} \cdot \mathrm{J}_{c_j c_i}.$$

Fig. 2-3 illustrates this pairwise camera constraint.

What do pairwise camera measurements tell us?

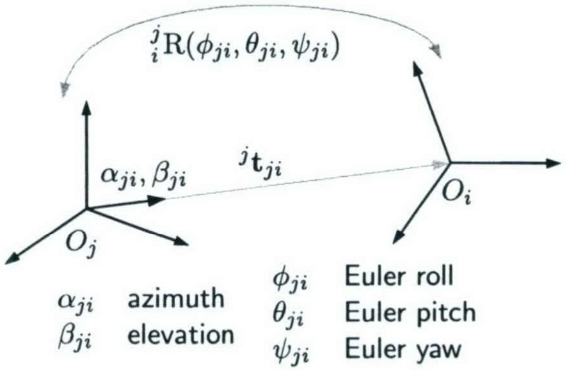
Now that we have derived how to model pairwise camera measurements, it's worth intuitively describing what equation (2.8) means in terms of reducing navigation error. First of all, pairwise camera measurements provide us with a bearing-only measurement of the baseline between poses — hence, we are dependent upon our navigation sensors to set the free-gauge scale. In our application this scale is implicitly fixed within the EKF by bounded

[2] Note that $\operatorname{atan2}(y,x) = \begin{cases} \tan^{-1}(y/x) & x > 0, y > 0 \\ \tan^{-1}(y/x) + \pi & x < 0, y > 0 \\ \tan^{-1}(y/x) & x > 0, y < 0 \\ \tan^{-1}(y/x) - \pi & x < 0, y < 0 \end{cases} = \tan^{-1}(y/x) + \text{constant}.$

This implies that $\frac{\partial \operatorname{atan2}(y,x)}{\partial(y,x)} \equiv \frac{\partial \tan^{-1}(y/x)}{\partial(y,x)}$, so that in general if $f = f(x)$ and $g = g(x)$, then

$$\frac{d \operatorname{atan2}(g,f)}{dx} = \frac{d \tan^{-1}(\frac{g}{f})}{dx} = \frac{1}{1 + (\frac{g}{f})^2} \underbrace{(\frac{dg}{dx} f^{-1} - g f^{-2} \frac{df}{dx})}_{\text{product-rule}} = \frac{1}{f^2 + g^2} (f \frac{dg}{dx} - g \frac{df}{dx}).$$

Figure 2-3 The pairwise 5-DOF camera measurement (i.e., relative-pose modulo scale).



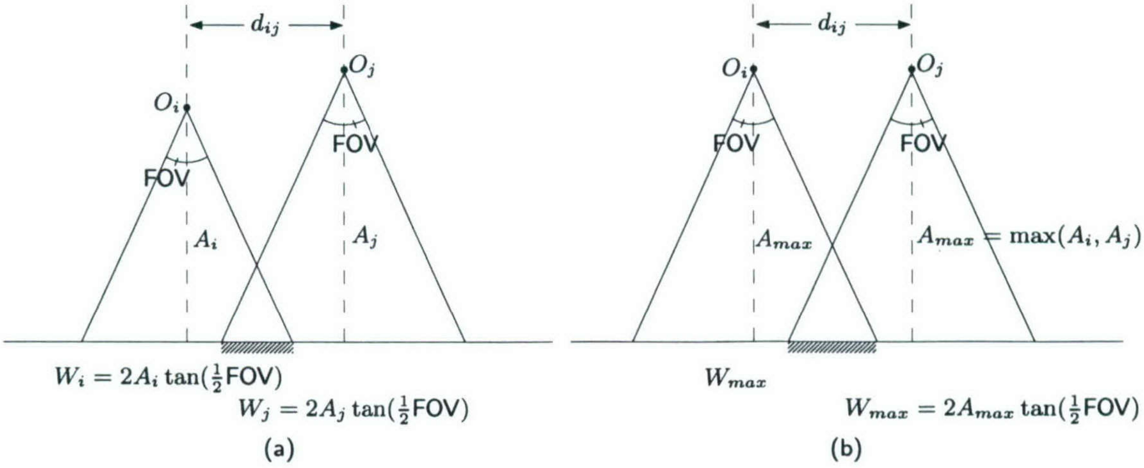
measurements of depth Z coming from a pressure sensor coupled with Doppler velocities that provide us with an integrated measurement of XYZ position. Secondly, (2.8) tells us that camera measurements can only reduce relative positional error components that are *orthogonal* to the baseline motion. Referring to Fig. 2-3 we see that frame O_i can slide anywhere along the baseline, ${}^{j}\mathbf{t}_{ji}$, without effecting our measure of azimuth/elevation. This suggests that temporal camera measurements do very little to reduce *along-track* error growth. Hence, long linear surveys will benefit far less from camera constraints than cross-over survey patterns, which have "loops" in their trajectory and result in ample spatial constraints. Finally, the nonlinear bearing-only constraints of (2.8) imply that linearization errors will be less significant in the EKF if we can maintain good map contact (e.g., typical grid-based surveys achieve this) to prevent our linearization point from "drifting" too far from the truth. This also suggests that when closing large loops, where the linearization point may be far from the true state, that we should incorporate the pairwise camera constraints in aggregate via some form of triangulation — a technique commonly used for feature-initialization in bearing-only SLAM applications [88].

2.3.3 Link Hypothesis

An essential task in a view-based representation is the hypothesis of *probable* overlapping image pairs. Since the image registration process is arguably the "slowest" component in the VAN framework, it is to our advantage to feed the registration module only likely candidate pairs so as to not waste time attempting registration on images that have a low probability of overlap. Our link hypothesis strategy is based upon a grossly-simplified 1D model for image overlap that uses our state estimate and measured scene altitude (beam-range measurements from the DVL) to project image footprints onto a horizontal plane as shown in Fig. 2-4. Since our imaging AUV flies in a closed-loop bottom-following mode for camera surveys, it approximately maintains a fixed altitude off of the seafloor. Therefore, for simplicity, we compute pairwise overlap using the larger altitude of a camera pair (Fig. 2-4(b)).

Assuming the above mentioned configuration, image percent overlap, ϵ, can be defined

Figure 2-4 Calculation of pairwise overlap for link hypothesis. To simplify the calculation of image overlap, we reduce it to a 1D case on a horizontal plane (a). In the illustrations below O_i and O_j are the camera centers, FOV is the field of view, A_i and A_j are the altimeter measured altitudes, W_i and W_j are the computed 1D image widths, and d_{ij} is the Euclidean baseline distance coming from our state estimate. Our vehicle approximately maintains a fixed altitude over the seafloor (this implies $A_i \approx A_j$), therefore, we simplify the calculation further by assuming the larger altitude for both cameras as shown in (b).



d_{ij}
O_i
O_j
FOV
FOV
A_i
A_j
$W_i = 2A_i \tan(\frac{1}{2}\text{FOV})$
$W_j = 2A_j \tan(\frac{1}{2}\text{FOV})$
(a)
d_{ij}
O_i
O_j
FOV
FOV
A_{max}
$A_{max} = \max(A_i, A_j)$
W_{max}
$W_{max} = 2A_{max} \tan(\frac{1}{2}\text{FOV})$
(b)

as

$$\epsilon = \begin{cases} 1 - \frac{d}{W_{max}} & 0 \leq d \leq W_{max} \\ 0 & \text{otherwise} \end{cases}.$$

Here, d is the Euclidean distance between the camera centers, $W_{max} = 2A_{max}\tan(\frac{1}{2}\text{FOV})$ is the 1D image width, A_{max} is the larger altitude of the pair, and FOV is the camera field of view. Under this scheme, we can set thresholds for minimum and maximum percent image overlap to obtain constraints on camera distance. We can then compute a first-order probability associated with whether or not the distance between the camera pair falls within these constraints. This calculation serves as the basis of our automatic link hypothesis algorithm, outlined in Algorithm 2.1, where all frames in our view-based map are checked to see whether or not they could overlap with the current robot view (i.e., linear complexity in the number of views). The k most likely candidates ($k = 5$ in our application) are then sent to our image registration module for comparison. While somewhat simplistic, we have obtained good results with this approximation and it has been the basis for the work presented in this thesis using automatically proposed links.

2.4 Generating the 5-DOF Camera Measurement

Having presented a view-based estimation framework capable of incorporating 5-DOF relative-pose measurements, we now turn our attention to explaining *how* we actually make the pairwise camera measurement. At the core is a feature-based image registration engine whose purpose is to generate pairwise measurements of relative-pose. Essential to this goal

Algorithm 2.1 View-based link hypothesis. Hypothesize which images I_i in our view-based map have a high probability of overlapping with the current robot view I_r.

```
define: $k$ {maximum number of candidates to return}
define: $\epsilon_{min} \in [0,1]$ {minimum percent overlap}
define: $\epsilon_{max} \in [0,1]$ {maximum percent overlap}
define: $\alpha \in [0,1]$ {confidence}
for all $I_i$ do
    $A_{max} \leftarrow \max(A_i, A_r)$
    $W_{max} \leftarrow 2A_{max} \tan(\frac{1}{2}\mathsf{FOV})$
    $d_{min} \leftarrow (1 - \epsilon_{max}) \cdot W_{max}$
    $d_{max} \leftarrow (1 - \epsilon_{min}) \cdot W_{max}$
    extract from our state $\boldsymbol{\xi}_t$ the joint-marginal $\begin{bmatrix} \mathbf{x}_{p_i} \\ \mathbf{x}_{p_r} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \boldsymbol{\mu}_{p_i} \\ \boldsymbol{\mu}_{p_r} \end{bmatrix}, \begin{bmatrix} \Sigma_{p_i p_i} & \Sigma_{p_i p_r} \\ \Sigma_{p_r p_i} & \Sigma_{p_r p_r} \end{bmatrix})$
    compute the relative camera pose $\mathbf{x}_{c_r c_i}$ and its first-order statistics (2.6),(2.7)
    using $\mathbf{x}_{c_r c_i}$ compute the Euclidean distance $d_i$ and its first-order statistics:
    $d_i \sim \mathcal{N}(\mu_{d_i}, \sigma^2_{d_i})$ where $d_i \leftarrow \|{}^{c_r}\mathbf{t}_{c_r c_i}\|$
    compute the probability $P_i$ that $d_{min} < d_i < d_{max}$:
    $P_i \leftarrow \int_{d_{min}}^{d_{max}} \mathcal{N}(\tau; \mu_{d_i}, \sigma^2_{d_i}) d\tau$
    if $P_i > \alpha$ then
        add $I_i$ to the candidate set $S$
    end if
end for
sort candidate set $S$ by $P_i$ and return up to the $k$ most probable candidates
```

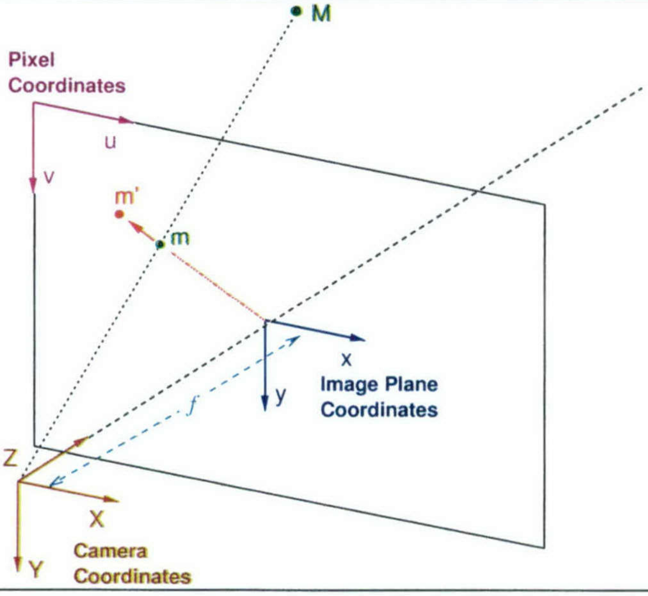
is the capability to cope with wide-baseline registration for two main reasons.

1. Low overlap imagery is common in our temporal image sequences due to the nature of our underwater application. Therefore, we must be able to deal with images in the temporal sequence having 35% or less overlap.

2. Loop-closing and cross-track spatial image constraints are the greatest strength of a VAN methodology. It is these measurements which help to correct dead-reckoned drift error and enforce recovery of a consistent trajectory. Since wide-baseline viewpoints are typical in this scenario, this condition would arise even if temporal overlap were much higher as with video-frame rates.

Thus, in order to be able to successfully handle wide-baseline image registration, our approach has been to extend a typical state-of-the-art feature-based image registration framework (§2.4.1) to judiciously exploit our navigation sensor capabilities wherever possible. For example, in §2.4.2 we show how we can exploit absolute orientation sensor measurements to reduce viewpoint variability in our feature encoding, and also obtain a good initialization for pairwise maximum likelihood refinement. We also show in §2.4.3 how we can use our pose prior and altitude measurements to improve the robustness of correspondence establishment via a novel pose-constrained correspondence search.

Figure 2-5 Illustration of a pinhole camera model. An intrinsically calibrated camera implies that the mapping from Euclidean camera coordinates to image pixel coordinates is known. The pinhole projective mapping from scene point **M** to image point **m** is described in homogeneous coordinates in terms of a 3×4 projection matrix $\mathrm{P} = \mathrm{K}[\mathrm{R}\,|\,\mathbf{t}]$ where K is the 3×3 upper triangular intrinsic parameter matrix and R,**t** describe the extrinsic coordinate transformation from scene to camera centered coordinates [64]. In practice we must also account for the lens distortion, which further maps **m** to **m′** [65].



2.4.1 Pairwise Feature-Based Image Registration

Calibrated Camera Model

Within our feature-based framework, we assume a standard calibrated pin-hole camera model [64] as illustrated in Fig. 2-5. This means that the homogeneous mapping from world to image plane can be described by a 3×4 projection matrix P defined as

$$\mathrm{P} = \mathrm{K}\left[{}^{c}_{w}\mathrm{R}\,|\,{}^{c}\mathbf{t}_{cw}\right]$$

where ${}^{c}_{w}\mathrm{R}$ and ${}^{c}\mathbf{t}_{cw}$ encode the coordinate transformation from world, w, to camera centered coordinate frame, c, and $\mathrm{K} = \left[\begin{smallmatrix} \alpha_u & s & u_o \\ 0 & \alpha_v & v_o \\ 0 & 0 & 1 \end{smallmatrix}\right]$ is the *known* 3×3 upper triangular intrinsic camera calibration matrix with α_u and α_v the pixel focal lengths in the x and y directions respectively, (u_o, v_o) is the principle point measured in pixels, and s is the pixel skew.

Under this representation the interest point with pixel coordinates (u, v) in image I is imaged as

$$\underline{\mathbf{u}} = \mathrm{P}\underline{\mathbf{X}} \tag{2.9}$$

where $\mathbf{u} = [u, v]^\top$ is the vector description of (u, v), $\underline{\mathbf{u}} = [\mathbf{u}^\top, 1]^\top$ its normalized homogeneous representation, $\mathbf{X} = [X, Y, Z]^\top$ is the imaged 3D scene point, and $\underline{\mathbf{X}} = [\mathbf{X}^\top, 1]^\top$ its normalized homogeneous representation. Note that for all homogeneous quantities, equality in expressions such as (2.9) is implicitly defined up to scale. The benefit of having a calibrated camera is that we can "undo" the projective mapping to pixel coordinates in (2.9) and instead work with normalized image plane coordinates as

$$\underline{\mathbf{x}} = \mathrm{K}^{-1}\underline{\mathbf{u}} = [{}^{c}_{w}\mathrm{R} \,|\, {}^{c}\mathbf{t}_{cw}]\underline{\mathbf{X}}.$$

The implication of this is that we can now describe the epipolar geometry in terms of the Essential matrix [64] and recover the 5-DOF camera pose from correspondences. For our application, we obtain the intrinsic calibration matrix K by calibrating in water using Zhang's planar method [180] and employ Heikkilä's radial/tangential distortion model [65] to compensate for both lens and index of refraction effects.

Geometric Feature-Based Algorithm

Our feature-based registration algorithm generally follows a state-of-the-art geometrical computer vision approach as described by Hartley and Zisserman [64] and Faugeras, Luong, and Papadopoulo [40]. Figures 2-6 and 2-7 illustrate the overall hierarchy of our feature-based algorithm founded on:

- Extract a combination of both Harris [63] and SIFT [93] interest points from each image. For the Harris points, we exploit our navigation prior to apply an orientation normalization to the interest regions by warping via the infinite homography [64], H_∞, and then compactly encode using Zernike moments [124].

- Establish putative correspondences between overlapping candidate image pairs based upon similarity and a pose-constrained correspondence search [36].

- Employ a statistically robust least median of squares (LMedS) [134] registration methodology with regularized sampling [179] to extract a consistent inlier correspondence set. For this task we use a 6-point Essential matrix algorithm [125] as the motion-model constraint.

- Solve for a relative-pose estimate using the inlier set and Horn's relative orientation algorithm [68] initialized with samples from our orientation prior (see §2.4.2).

- Carry out a two-view MLE refinement to extract the 5-DOF relative-pose constraint (i.e., azimuth, elevation, Euler roll, Euler pitch, Euler yaw) and first-order parameter covariance based upon minimizing the reprojection error over all inliers [64].

The remainder of this section focuses on the more novel aspects of the above approach. In particular, we discuss how to exploit sensor measured absolute orientation within a feature-based framework and in addition how we can use our state estimate to constrain correspondence searches.

2.4.2 Exploiting Sensor-Measured Absolute Orientation

Infinite Homography View Normalization

Establishing feature correspondences is arguably the most difficult task in a feature-based registration approach — this is especially true for wide-baseline registration. Without

Figure 2-6 An overview of the pairwise image registration engine. Dashed lines represent additional information provided by our state estimate, while bold boxes represent our systems-level extensions to a typical feature-based registration framework. Given two images I_i and I_j, we detect features using a combination of Harris and SIFT interest operators. For the Harris points, we exploit our navigation prior to orientation normalize the interest regions by warping via the infinite homography H_∞. For each feature, we establish a putative match based upon similarity and a novel pose-constrained correspondence search. A 6-point essential matrix algorithm employed within a statistically robust LMedS strategy extracts an inlier correspondence set. Using this set we initialize our relative-pose estimate using Horn's relative orientation algorithm with regularized sampling from our orientation prior and then refine in a two-view bundle adjustment step based upon minimizing the reprojection error over all inliers.

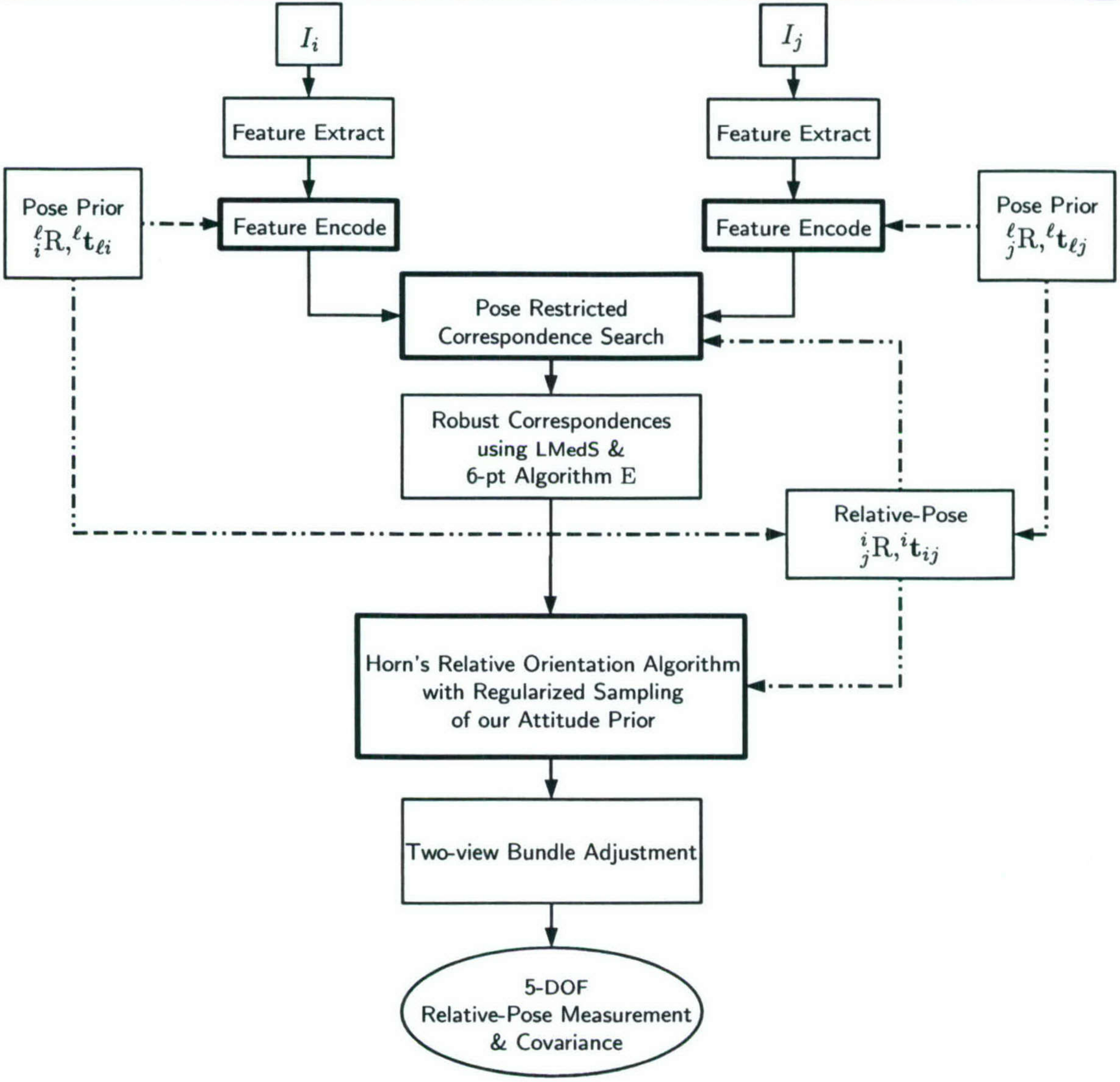
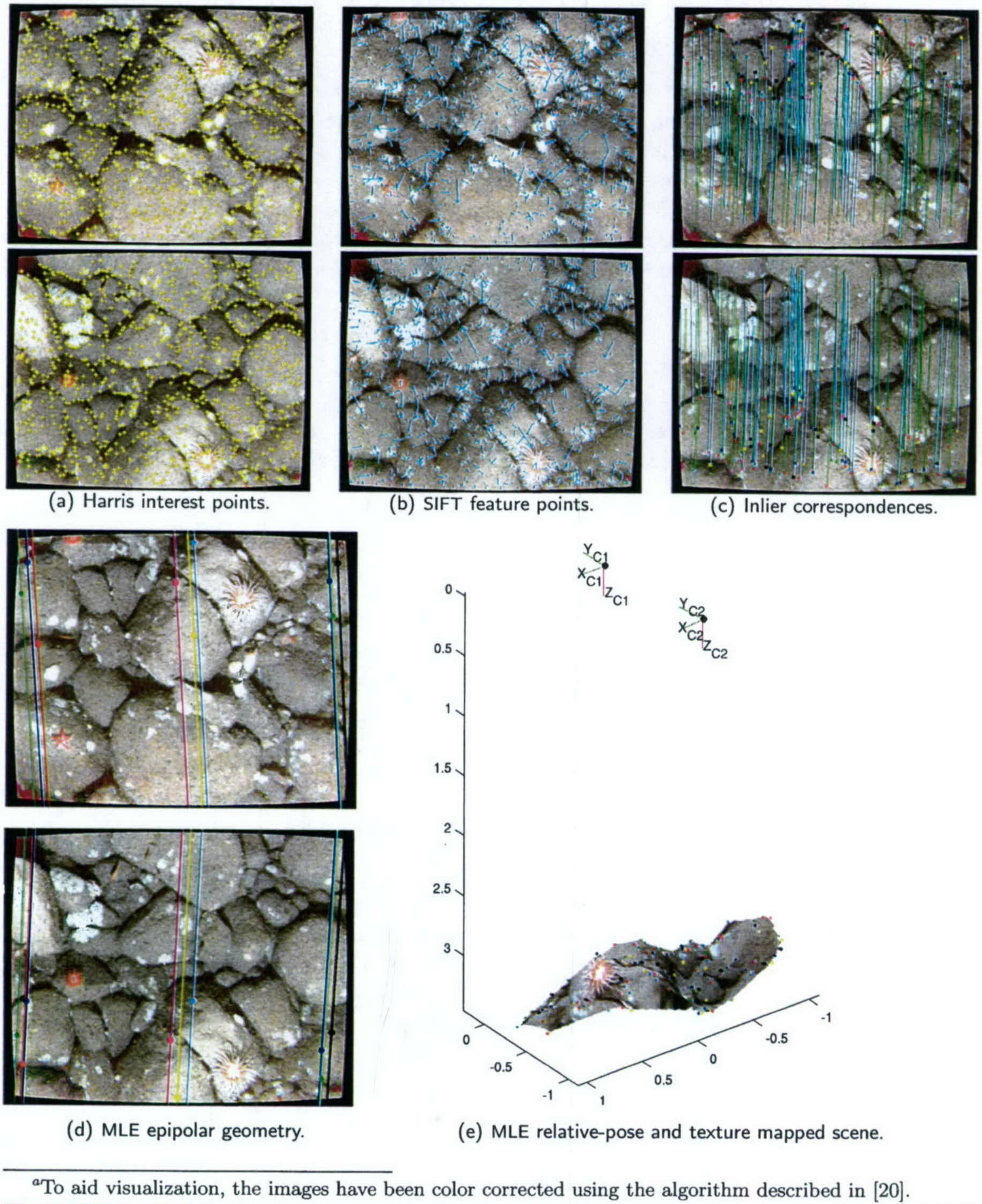

I_i
I_j
Feature Extract
Feature Extract
Pose Prior
${}^{\ell}_{i}\mathrm{R}, {}^{\ell}\mathbf{t}_{\ell i}$
Feature Encode
Feature Encode
Pose Prior
${}^{\ell}_{j}\mathrm{R}, {}^{\ell}\mathbf{t}_{\ell j}$
Pose Restricted Correspondence Search
Robust Correspondences using LMedS & 6-pt Algorithm E
Relative-Pose
${}^{i}_{j}\mathrm{R}, {}^{i}\mathbf{t}_{ij}$
Horn's Relative Orientation Algorithm with Regularized Sampling of our Attitude Prior
Two-view Bundle Adjustment
5-DOF Relative-Pose Measurement & Covariance

Figure 2-7 Typical output from the pairwise feature-based image registration module for a temporal pair of underwater images.[a] The pose and triangulated 3D feature points are the final product of a two-view MLE bundle adjustment step. The 3D triangulated feature points have been gridded in MATLAB to give a coarse surface approximation that has then been texture mapped with the common image overlap (the baseline magnitude is set to the navigation prior for visualization).



(a) Harris interest points.

(b) SIFT feature points.

(c) Inlier correspondences.

(d) MLE epipolar geometry.

(e) MLE relative-pose and texture mapped scene.

[a] To aid visualization, the images have been color corrected using the algorithm described in [20].

any knowledge of extrinsic camera information, robust techniques must rely upon encoding features in a viewpoint invariant way. For example, rotational and scale differences between images render simple correlation based similarity metrics useless. Therefore, to overcome these limitations, advanced techniques generally rely upon encoding some form of locally invariant feature descriptor such as differential invariants [140], generalized image moments [3,77,124], and affine invariant regions [106,139,164]. However, these higher-order descriptions also tend to be computationally expensive.

In the case of an instrumented platform with absolute measurements of orientation, we can utilize sensor-derived information to our advantage to relax the demands of the feature encoding while at the same time making it a more discriminatory metric. For example, in our application we use sensor-derived absolute pose information to prewarp the feature regions around our Harris interest points and precompensate for camera viewpoint orientation. Since attitude sensors provide information on the 3D orientation of cameras c and c' in a fixed reference frame w, we can normalize for orientation viewpoint effects via the infinite homography:

$$\mathrm{H}_\infty = \mathrm{K}\,{}^{c'}_{c}\mathrm{R}\,\mathrm{K}^{-1}.$$

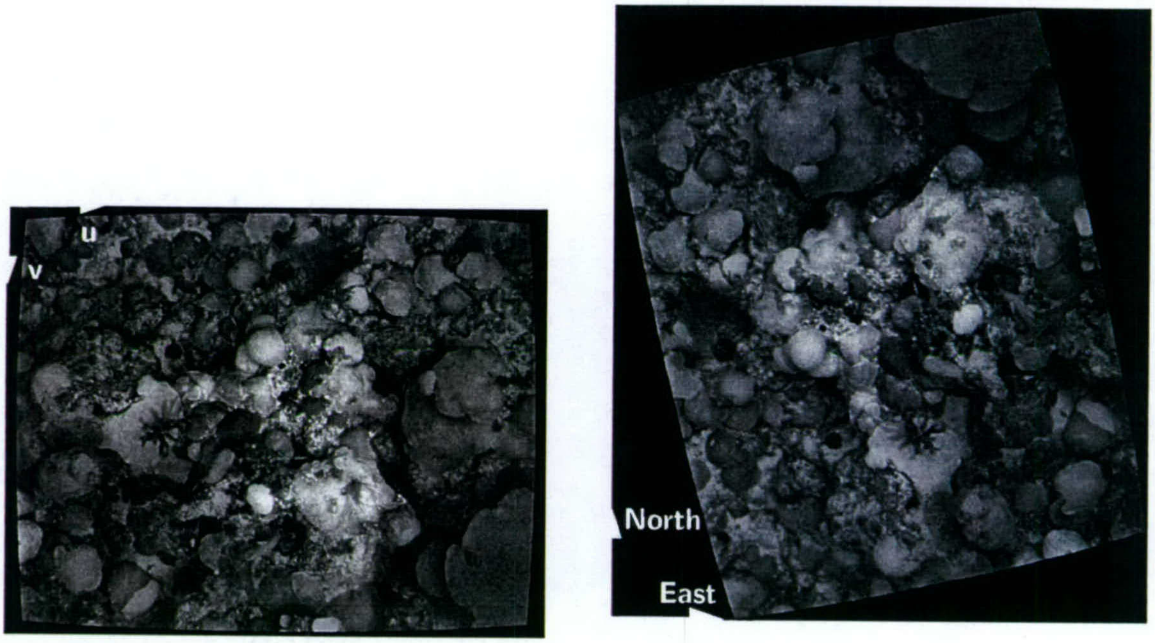
The infinite homography warps an image taken from camera orientation c into an image taken from orientation c' (note that the center of projection still remains at c). This viewpoint mapping is *exact* for points at *infinity* where $\underline{\mathbf{X}} = [X, Y, 0, 1]$, but otherwise can be used to compensate for viewpoint orientation (note that scene parallax is still present).

We compute H_∞ based upon our attitude estimate at image capture and apply it as an orientation correction to our images when encoding the Harris features. As demonstrated in Fig. 2-8, this warp effectively yields a synthetic view of the scene from a canonical camera coordinate frame aligned North, East, Down. This allows normalized correlation to be used as a similarity metric between Harris points and tends to work well for temporal image sequences by generating a high density of matches. This scheme in concert with SIFT features has proven to be successful for obtaining robust similarity matches.

Horn's Relative Orientation Algorithm and Samples from our Orientation Prior

We can also take advantage of our absolute orientation prior by obtaining an initial relative-pose solution using Horn's algorithm [68]. Given a set of inlier feature correspondences and an initial orientation guess, Horn's algorithm iteratively calculates a relative-pose estimate based upon enforcing the co-planarity condition over all ray pairs (i.e., if a ray from the left and right camera are to intersect then they must lie in a plane that also contains the baseline). If the orientation guess is approximately close to the true orientation, Horn's algorithm quickly converges to a minimal co-planarity error solution. Since orientation can be measured with bounded precision over the *entire* survey site while the camera baseline cannot, we use Horn's algorithm to obtain our initial 5-DOF relative-pose solution, which is then refined in a two-view bundle adjustment step based upon minimizing the reprojection error [64].

Figure 2-8 Synthetically normalizing the camera orientation via the infinite homography.



(a) Original distortion compensated coral image.

(b) Normalized image using H_∞ warp.

2.4.3 Pose-Constrained Correspondence Search

As mentioned in §2.4.2, the problem of initial feature correspondence establishment is arguably the most difficult and challenging task of a feature-based registration methodology. As we show in this section, having a pose prior relaxes the demands on the complexity of the feature descriptor — instead of having to be globally unique within an image, it now is required to be only locally unique. We use the epipolar geometry constraint expressed as a two-view point transfer model to restrict the correspondence search to probable *regions*. These regions are determined by our pose prior and are used to confine the interest point matching to a small subset of candidate correspondences. The benefit of this approach is that it simultaneously relaxes the demands of the feature descriptor while at the same time improves the robustness of similarity matching.

Epipolar Uncertainty Representation

Zhang [179] first characterized epipolar geometry uncertainty in terms of the covariance of the fundamental matrix while Shen [142] used knowledge of the pose prior to restrict the correspondence search to bands along the epipolar line calculated by propagating pose uncertainty. However, a criticism of both of these characterizations is that the uncertainty representation is hard to interpret in terms of physical parameters — how does one interpret the covariance of a line? Our approach is to use a two-view point transfer mapping that benefits from a direct physical interpretation of the pose parameters and in addition can take advantage of scene range data if available. While similar to Lanser's technique [83], our approach does not assume or require that an *a priori* CAD model of the environment exist.

Two-View Point Transfer Model

In deriving the point transfer mapping we assume projective camera matrices $\mathrm{P} = \mathrm{K}[\mathrm{I}\,|\,\mathbf{0}]$ and $\mathrm{P}' = \mathrm{K}[\mathrm{R}\,|\,\mathbf{t}]$ where for notational convenience we drop the explicit subscript/superscript notation and simply write the relative orientation parameters as R,$\mathbf{t}$ (i.e., $\mathrm{R} = {}^{c'}_{c}\mathrm{R}$ and ${}^{c'}\mathbf{t}_{c'c}$). We begin by noting that the scene point $\mathbf{X}$ is projected through camera P as

$$\underline{\mathbf{u}} = \mathrm{P}\underline{\mathbf{X}} = \mathrm{K}\mathbf{X},$$

which implies that explicitly accounting for scale we have

$$\mathbf{X} \equiv Z\mathrm{K}^{-1}\underline{\mathbf{u}}. \tag{2.10}$$

The back-projected scene point $\mathbf{X}$ can subsequently be reprojected into image I' as

$$\underline{\mathbf{u}}' = \mathrm{P}'\underline{\mathbf{X}} = \mathrm{K}(\mathrm{R}\mathbf{X} + \mathbf{t}). \tag{2.11}$$

By substituting (2.10) into (2.11) and recognizing that the following relation is up to scale, we obtain the homogeneous point transfer mapping [64]:

$$\underline{\mathbf{u}}' = \mathrm{KRK}^{-1}\underline{\mathbf{u}} + \mathrm{K}\mathbf{t}/Z. \tag{2.12}$$

Finally, by explicitly normalizing (2.12) we recover the *non-homogeneous* point transfer mapping

$$\mathbf{u}' = \frac{\mathrm{H}_\infty\underline{\mathbf{u}} + \mathrm{K}\mathbf{t}/Z}{\mathbf{H}_\infty^{3\top}\underline{\mathbf{u}} + t_z/Z} \tag{2.13}$$

where $\mathrm{H}_\infty = \mathrm{KRK}^{-1}$, $\mathbf{H}_\infty^{3\top}$ refers to the third row of H_∞, and t_z is the third element of $\mathbf{t}$.

When the depth of the scene point Z is known in camera frame c, then (2.13) describes the exact two-view point transfer mapping. However, when Z is unknown, then (2.13) describes a functional relationship on Z (i.e., $\mathbf{u}' = f(\mathbf{u}, Z)$) that traces out the corresponding epipolar line in I'.

Proof. Note that if $\underline{\mathbf{u}}'$ lies on the epipolar line l' then it must satisfy

$$\underline{\mathbf{u}}'^\top \mathrm{l}' = 0. \tag{2.14}$$

Thus, if we can show that for all values of Z, (2.12) satisfies (2.14), then we can deduce (2.13) parameterizes the epipolar line in I' as a function of Z.[3]

$$\begin{aligned} \underline{\mathbf{u}}'^\top \mathrm{l}' &= 0 \\ \underline{\mathbf{u}}'^\top \mathrm{F}\underline{\mathbf{u}} &= 0 \quad \mathrm{l}' \equiv \mathrm{F}\mathbf{u}, \text{ where F is the fundamental matrix} \\ \underline{\mathbf{u}}'^\top \underbrace{\mathrm{K}^{-\top}[\mathbf{t}]_\times \mathrm{RK}^{-1}}_{\mathrm{F}}\underline{\mathbf{u}} &= 0 \quad \text{expanding F} \end{aligned}$$

$$
\begin{aligned}
\underbrace{\left(\mathrm{KRK}^{-1}\underline{\mathbf{u}}+\mathrm{K}\mathbf{t}/Z\right)}_{\underline{\mathbf{u}}'}{}^{\top}\mathrm{K}^{-\top}[\mathbf{t}]_{\times}\mathrm{RK}^{-1}\underline{\mathbf{u}}&=0 \quad \text{substituting (2.12)}\\
\left(\mathrm{K}\boldsymbol{\alpha}+\mathrm{K}\mathbf{t}/Z\right)^{\top}\mathrm{K}^{-\top}[\mathbf{t}]_{\times}\boldsymbol{\alpha}&=0 \quad \text{defining } \boldsymbol{\alpha}=\mathrm{RK}^{-1}\underline{\mathbf{u}}\\
\left(\boldsymbol{\alpha}^{\top}\mathrm{K}^{\top}+\mathbf{t}^{\top}\mathrm{K}^{\top}/Z\right)\mathrm{K}^{-\top}[\mathbf{t}]_{\times}\boldsymbol{\alpha}&=0\\
\left(\boldsymbol{\alpha}^{\top}+\mathbf{t}^{\top}/Z\right)[\mathbf{t}]_{\times}\boldsymbol{\alpha}&=0\\
\boldsymbol{\alpha}\cdot(\mathbf{t}\times\boldsymbol{\alpha})+\tfrac{1}{Z}\mathbf{t}\cdot(\mathbf{t}\times\boldsymbol{\alpha})&=0\\
0&=0 \quad \forall \text{ values } Z \qquad \text{Q.E.D.}
\end{aligned}
$$

Point Transfer Mapping with Uncertainty

Now that we've derived the two-view point transfer mapping (2.13), in this section we show how we can use it to constrain our correspondence search between image pair I_i, I_j by using our pose prior knowledge from $\boldsymbol{\xi}_t$. We begin by defining the parameter vector $\boldsymbol{\gamma}$ as

$$
\boldsymbol{\gamma}=[\mathbf{x}_{p_i}^{\top},\mathbf{x}_{p_j}^{\top},Z,u,v]^{\top} \tag{2.15}
$$

with mean $\boldsymbol{\mu}_{\gamma}$ and covariance Σ_{γ} given by

$$
\boldsymbol{\mu}_{\gamma}=\begin{bmatrix}\boldsymbol{\mu}_{p_i}\\ \boldsymbol{\mu}_{p_j}\\ Z\\ u\\ v\end{bmatrix} \qquad \Sigma_{\gamma}=\begin{bmatrix}\Sigma_{p_ip_i} & \Sigma_{p_ip_j} & 0 & 0 & 0\\ \Sigma_{p_jp_i} & \Sigma_{p_jp_j} & 0 & 0 & 0\\ 0 & 0 & \sigma_Z^2 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}.
$$

Here, $\mathbf{x}_{p_i}$, $\mathbf{x}_{p_j}$ are the delayed-state vehicle poses from $\boldsymbol{\xi}_t$ used to calculate the relative camera pose $\mathbf{x}_{c_jc_i}$ (2.6), Z and σ_Z represent the scene depth parameters as measured in camera frame i, and (u,v) describe the feature location in pixels in image I_i. Note that in defining Σ_{γ} we employ the common assumption that features are extracted with isotropic, independent, unit variance noise [64]. To obtain a first-order estimate of the uncertainty in the point transfer mapping between I_i and I_j we compute

$$
\boldsymbol{\mu}_{u'}\approx(2.13)\big|_{\boldsymbol{\mu}_{\gamma}} \tag{2.16}
$$

$$
\Sigma_{u'}\approx\mathrm{J}\Sigma_{\gamma}\mathrm{J}^{\top} \tag{2.17}
$$

where $\boldsymbol{\mu}_{u'}$ is the predicted point location of $\mathbf{u}$ in I_j, $\Sigma_{u'}$ its variance, and $\mathrm{J}=\frac{\partial\mathbf{u}'}{\partial\boldsymbol{\gamma}}$ is the point transfer Jacobian.[4] We use the Gaussian distribution as an analytical tool to compute first-order search bounds in (u',v') space by noting that

$$
\left(\mathbf{u}'-\boldsymbol{\mu}_{u'}\right)^{\top}\Sigma_{u'}^{-1}\left(\mathbf{u}'-\boldsymbol{\mu}_{u'}\right)=k^2 \tag{2.18}
$$

[3] $[\mathbf{a}]_{\times}$ denotes a skew symmetric matrix implementing the vector cross-product (i.e., $[\mathbf{a}]_{\times}\mathbf{b}\equiv\mathbf{a}\times\mathbf{b}$).

[4] We compute this Jacobian numerically using the algorithm described in [64, §A4.2].

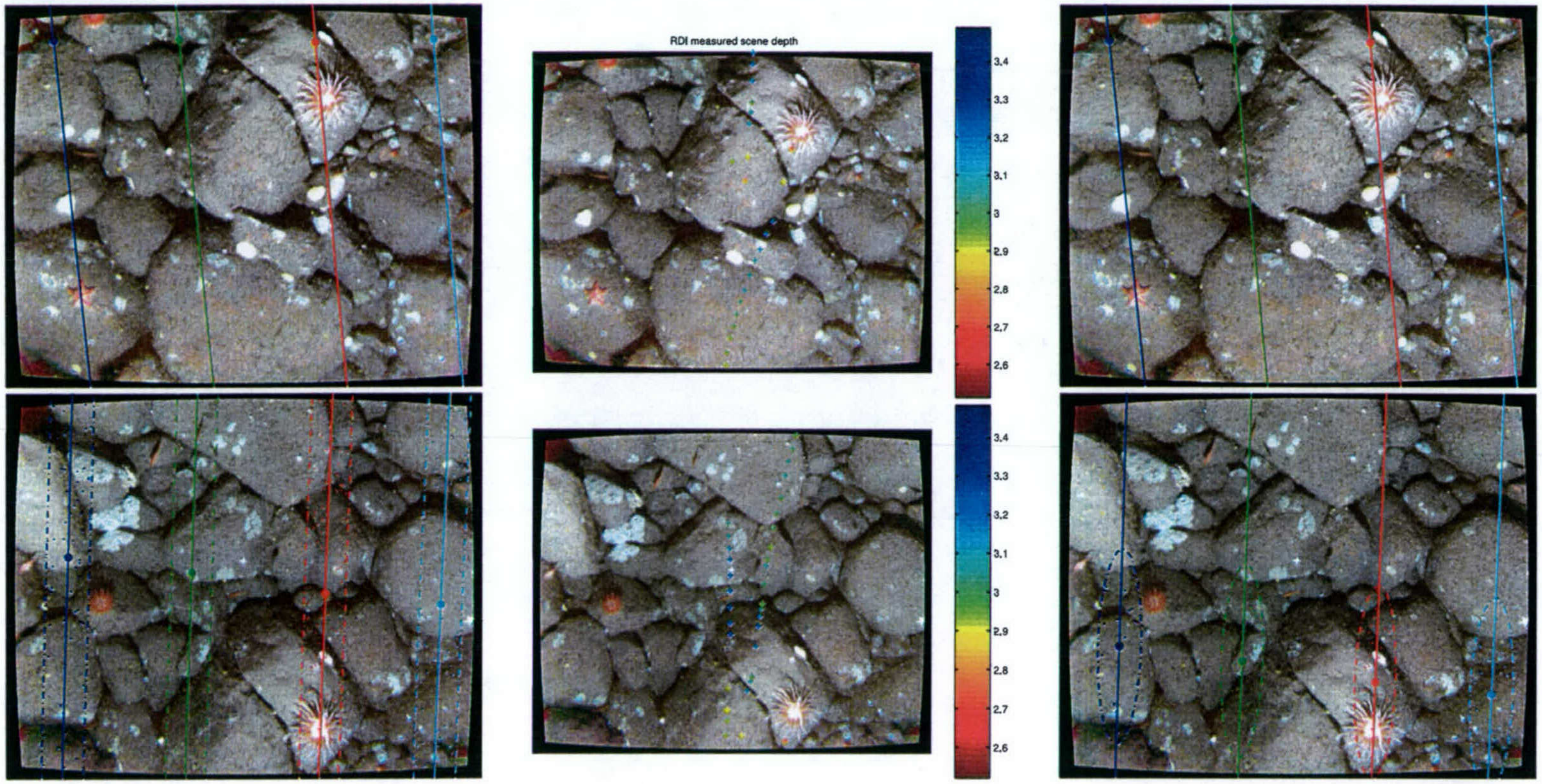
defines an ellipse where k^2 follows a χ^2_2 distribution. Hence, we can choose an appropriate k^2 such that with probability α the true mapping $\mathbf{u}'_o$ falls within this region. Under this scheme we test all feature points in I_j to see if they fall within the ellipse (2.18), and if they do, then they are considered to be candidate matches for $\mathbf{u}$. Since relative-pose uncertainty depends on the reference frame in which it is expressed, we apply the two-view search constraint both forwards and backwards to obtain a consistent candidate correspondence set. In other words, candidate matches in I_j that correspond to interest points in I_i are checked to see if they map back to the generating interest point in I_i. Based upon this set of consistent candidate matches, feature similarity is then used to establish the one-to-one putative correspondence set.

Algorithm 2.2 describes the pose-constrained correspondence search in pseudo-code where we use scene depth, Z, and its uncertainty, σ_Z, as a convenient parameterization for controlling the size and shape of the search regions in I_j as illustrated in Fig. 2-9. For example, in the case where no *a priori* knowledge of scene depth is available, choosing any finite value for Z and setting $\sigma_Z \rightarrow \infty$ recovers a search *band* along the epipolar line in I_j whose width corresponds to the uncertainty in relative camera pose, $\mathbf{x}_{c_j c_i}$ (Fig. 2-9(a)). On the other hand, when knowledge of an average scene depth, Z_{avg}, exists (e.g., from an altimeter) (Fig. 2-9(b)), then it and an appropriately chosen σ_Z can be used to limit the search space to *ellipses* centered along the epipolar lines (Fig. 2-9(c)). Furthermore, in the case where dense scene range measurements are available (e.g., from a laser range finder or scanning pencil-beam sonar), then scene depth, Z, can be assigned on a point-by-point basis with high precision. In any case, the pose-constrained correspondence search greatly improves the reliability and robustness of feature similarity matching by reducing the candidate correspondence set to a relatively few number of options as demonstrated in Fig. 2-10.

Algorithm 2.2 Pose-constrained correspondence search.

```
Require: $U_i$, $U_j$, $\boldsymbol{\mu}_{p_i}$, $\boldsymbol{\mu}_{p_j}$, $\Sigma_{p_i p_i}$, $\Sigma_{p_i p_j}$, $\Sigma_{p_j p_j}$, $Z$, $\sigma_Z^2$
    {$U_i$, $U_j$ are the set of feature points in $I_i$, $I_j$ respectively}
$C_{ij} \leftarrow 0_{U_i \times U_j}$ {initialize feature correspondence matrix $ij$ to all zeros}
for all $\mathbf{u}_i \in U_i$ do {forward mapping from $I_i$ to $I_j$}
    assemble $\boldsymbol{\gamma}$ as in (2.15)
    do point transfer $\boldsymbol{\mu}_{u'_i} \leftarrow (2.16)\big|_{\boldsymbol{\mu}_\gamma}$   $\Sigma_{u'_i} \leftarrow (2.17)\big|_{\Sigma_\gamma}$
    for all $\mathbf{u}_j \in U_j$ do {test all points in $I_j$}
        if $(\mathbf{u}_j - \boldsymbol{\mu}_{u'_i})^\top \Sigma^{-1}_{u'_i} (\mathbf{u}_j - \boldsymbol{\mu}_{u'_i}) < k^2$ then {$\mathbf{u}_j$ lies in the ellipse}
            $C_{ij}(\mathbf{u}_i, \mathbf{u}_j) \leftarrow 1$ {flag $\mathbf{u}_i$, $\mathbf{u}_j$ as candidate match}
        end if
    end for
end for
repeat the above with the roles of $U_i$, $U_j$ reversed to get $C_{ji}$
$C \leftarrow C_{ij} \cap C_{ji}^\top$ {forwards/backwards intersection yields a consistent candidate set}
assign putative matches based upon similarity measure within the restricted set $C$
```

Figure 2-9 Pose-constrained correspondence regions for a temporal pair of underwater images arranged I_i above and I_j below; the two-view mapping is shown for $I_i \rightarrow I_j$. (a) Color-coded pose instantiated epipolar lines are shown in both I_i and I_j. The search *bands* in I_j correspond to *no* knowledge of scene depth with width attributable to relative-pose uncertainty. (b) Altimeter measured scene depth projected into the image plane of each view (the altimetry is derived from the beam range measurements of the DVL). (c) The search regions now become *ellipses* based upon the altimetry constraint.

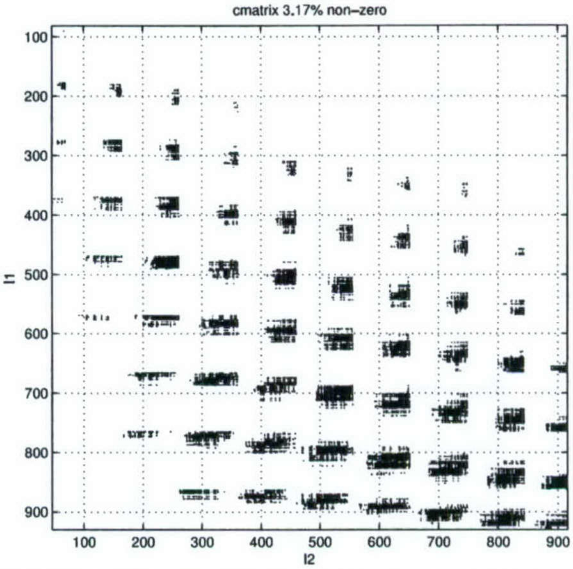


(a) Search bands corresponding to no scene depth prior.

(b) Altimeter measurements of scene depth in meters projected into image plane.

(c) Search bands become regions based upon altimeter measured scene depth constraint.

Figure 2-10 The pose-constrained candidate correspondence matrix for Fig. 2-9(c). The rows/columns correspond to a ordering of the feature indices in I_i/I_j respectively, where a nonzero entry indicates a potential match. Note that without any *a priori* pose knowledge this matrix would be full. Hence, we would be forced to rely purely upon the discriminatory power of the feature similarity measure to establish correspondences. Below, we see that the pose-restricted search constraint has reduced the possible space of matches by over 97%.



2.4.4 Are Pairwise Camera Measurements Correlated?

We now address the question of whether or not pairwise camera measurements are correlated. Recall that a primary assumption in an EKF fusion framework is that measurements are assumed to be corrupted by time independent noise (2.1).[5] In our view-based framework, raw images are registered directly to produce a relative-pose measurement that is then fed to the EKF filter as a relative observation between two states. If a raw image is used multiple times to make multiple measurements, for example $I_i \leftrightarrow I_j$ and $I_j \leftrightarrow I_k$, then this raises the possibility that camera measurements $\mathbf{z}_{ij}$ and $\mathbf{z}_{jk}$ may be correlated. Neglecting such a correlation would put too much weight on the measurement update step since it would treat each observation as an *independent* corroboration of the other. Unfortunately, as with any view-based "scan-matching" framework that "reuses" raw data, actually computing the measurement correlation is intractable and, thus, out of practicality measurements are assumed independent [61, 94]. However, in the case of measurements made from low-overlap imagery, we argue that this independence assumption is not particularly far from the truth.

Our camera-derived relative-pose measurement and covariance are generated as an end-product of a feature-based two-view maximum likelihood estimate based upon minimizing

[5] A typical strategy for dealing with time-correlated measurements is to augment the state description with an appropriately chosen linear system driven by white noise. The output of this "coloring" process is then added to an otherwise noiseless observation model to account for the time-correlated measurement noise [54].

the reprojection error. As is common in the vision community, the image feature locations are assumed to be corrupted by independent isotropic noise of unit variance [64] (measured in pixels). Denoting the set of common features between $I_i \leftrightarrow I_j$ as F_{ij}, and the set between $I_j \leftrightarrow I_k$ as F_{jk}, the implication of this noise assumption is that for null pairwise feature intersection (i.e., $F_{ij} \cap F_{jk} = \varnothing$), the corresponding camera measurements $\mathbf{z}_{ij}$ and $\mathbf{z}_{jk}$ are *uncorrelated.* Hence, assuming pairwise independence is true for temporally consecutive images with 50% or less overlap, and *approximately* true for spatial measurements where the number of re-observed point correspondences is low.[6]

Proof. Assume three camera frames $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ where image pairs (I_1, I_2) and (I_2, I_3) have spatial overlap, but (I_1, I_3) do not. Define $\mathcal{C}_2$ to be the reference coordinate frame and $\mathbf{X}$ to be the set of 3D world points viewed by the three camera system (i.e., $\mathbf{X} = \{\mathbf{X}_{12}, \mathbf{X}_{23}\}$ where $\mathbf{X}_{ij}$ is the set of 3D points viewed by camera $\mathcal{C}_i$ and $\mathcal{C}_j$ and $\mathbf{X}_{13} = \{\varnothing\}$).

For the two-view MLE bundle adjustment we define our parameter vector $\mathbf{q}$ to be

$$\mathbf{q} = \left[\mathbf{p}_{21}^\top, \mathbf{p}_{23}^\top, \mathbf{X}^\top\right]^\top$$

where $\mathbf{p}_{21}$ and $\mathbf{p}_{23}$ represent the pose vectors of cameras $\mathcal{C}_1$ and $\mathcal{C}_3$ with respect to camera frame $\mathcal{C}_2$ respectively. The reprojection error is defined as

$$\begin{aligned}
\mathbf{e}_{1_i} &= \mathbf{u}_{1_i} - \mathrm{P}(\mathrm{K}, \mathbf{p}_{21}, \underline{\mathbf{X}}_{12_i}) && \text{Camera 1} \\
\mathbf{e}_{2_i} &= \mathbf{u}_{2_i} - \mathrm{P}(\mathrm{K}, 0_{6\times 1}, \underline{\mathbf{X}}_{i}) && \text{Camera 2} \\
\mathbf{e}_{3_i} &= \mathbf{u}_{3_i} - \mathrm{P}(\mathrm{K}, \mathbf{p}_{23}, \underline{\mathbf{X}}_{23_i}) && \text{Camera 3}
\end{aligned}$$

where $\mathbf{u}_{n_i}$ is the i^{th} feature point measured in image I_n, K is the camera calibration matrix, $\mathbf{X}_i$ is the i^{th} 3D point, and $\mathrm{P}(\mathrm{K}, \mathbf{p}_{nm}, \mathbf{X}_{nm_i})$ denotes the pinhole projection of the 3D point into the image plane. Stacking all of the reprojection error equations we have

$$\boldsymbol{\varepsilon}_{\text{total}} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \boldsymbol{\varepsilon}_3 \end{bmatrix}$$

where $\boldsymbol{\varepsilon}_1$, $\boldsymbol{\varepsilon}_2$, $\boldsymbol{\varepsilon}_3$ are the individual image error measurements.

$$\boldsymbol{\varepsilon}_1 = [\mathbf{e}_{1_1}^\top, \cdots, \mathbf{e}_{1_m}^\top]^\top \qquad \boldsymbol{\varepsilon}_2 = [\mathbf{e}_{2_1}^\top, \cdots, \mathbf{e}_{2_{m+n}}^\top]^\top \qquad \boldsymbol{\varepsilon}_3 = [\mathbf{e}_{3_1}^\top, \cdots, \mathbf{e}_{3_n}^\top]^\top$$

The goal of bundle adjustment is to minimize the total squared error cost

$$C = \boldsymbol{\varepsilon}_{\text{total}}^\top \boldsymbol{\varepsilon}_{\text{total}}$$

over all views by optimizing over camera poses and scene structure and is considered to be the gold-standard by which all other measures are judged. This nonlinear optimization problem is solved from an initial guess via a large-scale, sparse, partitioned Levenberg-Marquardt algorithm [64,128] that takes advantage of the sparse reprojection error Jacobian,

[6] Note that both of these criteria are usually met in a typical grid-based AUV survey where both cross-track and temporal overlap are of the order of 20–35%.

which for the problem under consideration is

$$\mathrm{J} = \frac{\partial \varepsilon_{\text{total}}}{\partial \mathbf{q}} = \left[\frac{\partial \varepsilon_{\text{total}}}{\partial \mathbf{p}_{21}}, \frac{\partial \varepsilon_{\text{total}}}{\partial \mathbf{p}_{23}}, \frac{\partial \varepsilon_{\text{total}}}{\partial \mathbf{X}_{12}}, \frac{\partial \varepsilon_{\text{total}}}{\partial \mathbf{X}_{23}} \right] = \begin{bmatrix} \frac{\partial \varepsilon_1}{\partial \mathbf{p}_{21}} & 0_{m\times 6} & \frac{\partial \varepsilon_1}{\partial \mathbf{X}_{12}} & 0_{m\times n} \\ 0_{(m+n)\times 6} & 0_{(m+n)\times 6} & \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{12}} & \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{23}} \\ 0_{n\times 6} & \frac{\partial \varepsilon_3}{\partial \mathbf{p}_{23}} & 0_{n\times m} & \frac{\partial \varepsilon_3}{\partial \mathbf{X}_{23}} \end{bmatrix}.$$

The MLE estimate $\hat{\mathbf{q}}$ corresponds to the local minima of C with a first-order estimate of its parameter covariance given by [64]

$$\begin{aligned} \Sigma_{\hat{q}} &= \mathrm{J}^\top \mathrm{J} \\ &= \left[\begin{array}{cc|cc} \frac{\partial \varepsilon_1}{\partial \mathbf{p}_{21}}^\top \frac{\partial \varepsilon_1}{\partial \mathbf{p}_{21}} & 0_{6\times 6} & \frac{\partial \varepsilon_1}{\partial \mathbf{p}_{21}}^\top \frac{\partial \varepsilon_1}{\partial \mathbf{X}_{21}} & 0_{6\times m} \\ 0_{6\times 6} & \frac{\partial \varepsilon_3}{\partial \mathbf{p}_{23}}^\top \frac{\partial \varepsilon_3}{\partial \mathbf{p}_{23}} & 0_{6\times m} & \frac{\partial \varepsilon_1}{\partial \mathbf{p}_{23}}^\top \frac{\partial \varepsilon_1}{\partial \mathbf{X}_{23}} \\ \hline \frac{\partial \varepsilon_1}{\partial \mathbf{X}_{12}}^\top \frac{\partial \varepsilon_1}{\partial \mathbf{p}_{21}} & 0_{m\times 6} & \frac{\partial \varepsilon_1}{\partial \mathbf{X}_{12}}^\top \frac{\partial \varepsilon_1}{\partial \mathbf{X}_{12}} + \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{12}}^\top \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{12}} & \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{12}}^\top \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{23}} \\ 0_{6\times 6} & \frac{\partial \varepsilon_3}{\partial \mathbf{X}_{23}}^\top \frac{\partial \varepsilon_3}{\partial \mathbf{p}_{23}} & \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{23}}^\top \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{12}} & \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{23}}^\top \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{23}} + \frac{\partial \varepsilon_3}{\partial \mathbf{X}_{23}}^\top \frac{\partial \varepsilon_3}{\partial \mathbf{X}_{23}} \end{array} \right]. \end{aligned}$$

Note that in the above, we've employed the common assumption of isotropic, independent, unit variance feature pixel noise. To make interpretation easier, we've partitioned the parameter covariance $\Sigma_{\hat{q}}$ into camera poses (upper-left) and 3D structure (lower-right). Careful inspection of $\Sigma_{\hat{q}}$ clearly shows that for null feature intersection, the two relative camera poses $\mathbf{p}_{21}$ and $\mathbf{p}_{23}$ are uncorrelated.[7] Q.E.D.

2.5 Results

In this section we present results demonstrating VAN's application to underwater pose estimation. The first set of results is for a real-world dataset collected by the SeaBED AUV during a benthic habitat classification survey conducted at the Stellwagen Bank National Marine Sanctuary. The second set of results are for experimental validation of the VAN framework using a ROV at the Johns Hopkins University (JHU) Hydrodynamic Test Facility with ground-truth.

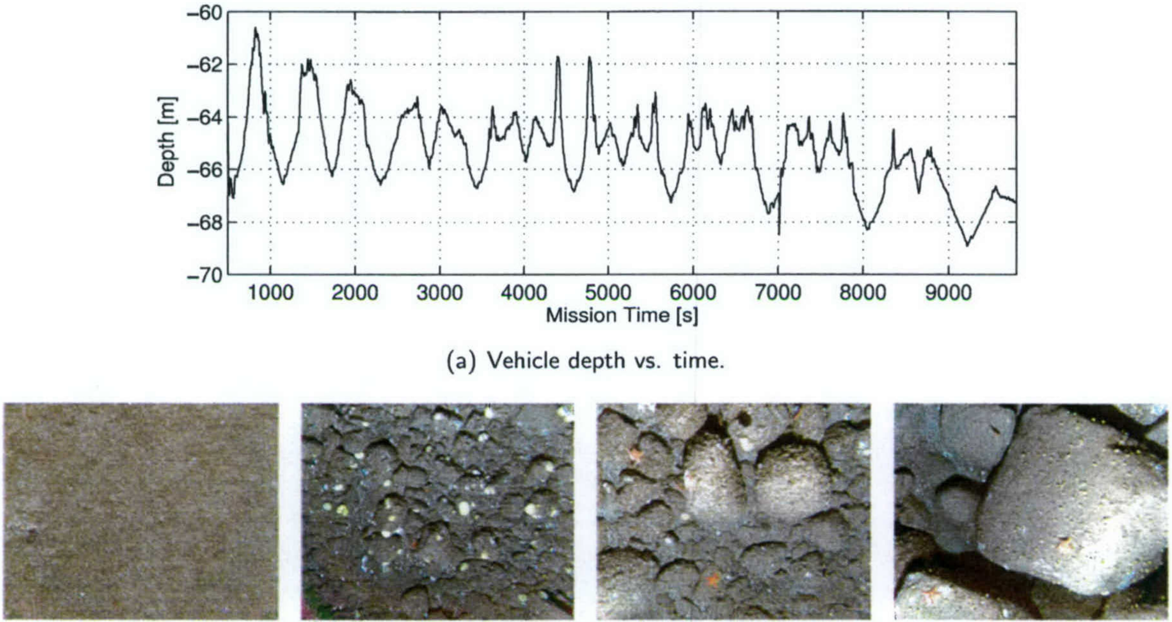
2.5.1 Real-World Results: Stellwagen Bank

Experimental Setup

The SeaBED AUV conducted a grid-based survey for a portion of the Stellwagen Bank National Marine Sanctuary in March 2003 [145]. The vehicle is equipped with a single down-looking camera and is instrumented with the navigation sensor suite tabulated in Table 2.1, pg. 50. As depicted in Fig. 2-11, SeaBED conducted the survey in a bottom-following mode where it tried to maintain constant altitude over a sloping rocky ocean bottom. The intended survey pattern consisted of 15 North/South legs each 180 meters long and spaced 1.5 meters apart while maintaining an average altitude of 3.0 meters above

[7] Also note that the 3D structure $\mathbf{X}_{12}$ and $\mathbf{X}_{23}$ are uncorrelated as well since $\frac{\partial \varepsilon_2}{\partial \mathbf{X}_{12}}^\top \frac{\partial \varepsilon_2}{\partial \mathbf{X}_{23}} = 0_{m\times n}$

Figure 2-11 A depiction of Stellwagen Bank's topography. Since the vehicle was trying to maintain constant-altitude the depth plot (a) is a proxy for terrain variation — notice that the depth excursions are on the order of several meters.



(a) Vehicle depth vs. time.

(b) Bottom topography ranges from sand to boulders.

the seafloor at a forward velocity of 0.35 m/s. Closed-loop feedback on the DR navigation estimate was used for real-time vehicle control.

We processed a small subset of the dataset using 100 images from a South/North trackline pair, the results of which are shown in Fig. 2-13. Plot (b) on the right depicts the VAN estimated camera trajectory and its 3-sigma confidence bounds. Successfully registered image pairs are indicated by the red and green links connecting the camera poses where green corresponds to temporally consecutive image frames and red to spatially neighboring image frames. For comparison purposes, plot (a) on the left depicts the DR trajectory overlaid on top of the VAN estimated XY trajectory. Both plots are in meters where X is East and Y is North.

Our feature-based registration algorithm was successful in automatically establishing putative correspondences between sequential image pairs (green links), however, automatic cross-track image registration (red links) proved to be too difficult. The cause for this was due to significant variation in scene appearance as illuminated from different vantage points.[8] Furthermore, these variations were complicated by the fact that the vehicle flew on reciprocal headings alternating every other leg of the grid-based survey. Hence, shadows were cast in opposite directions for parallel tracklines. Therefore, for this dataset cross-track putative correspondences were manually established for 19 image pairs, which are indicated by the red spatial links in Fig. 2-13.

[8] The example shown in Fig. 1-13(b), pg. 36 displays cross-track imagery from this dataset.

Experimental Results

A number of important observations in Fig. 2-13 are worth pointing out.

1. First, note that the VAN uncertainty ellipses are smaller for camera poses that are constrained by spatial links. Since spatial links provide a mechanism for relating past vehicle poses to the present, they also provide a means for correcting DR drift error. While trajectory uncertainty in a DR navigation system grows monotonically unbounded with time, in contrast VAN's error growth is essentially a function of network topology and distance away from the zeroth reference node (i.e., the first image).

2. A second observation worth noting is the delayed-state smoothing that occurs in the VAN framework. Spatial links not only decrease the uncertainty of the image pair involved, but also decrease the uncertainty of other delayed-states that are correlated. In particular, Fig. 2-12 characterizes the time-evolution of the view-based map uncertainty by plotting the trace of the XY sub-block versus image frame number. Note the sudden decrease occurring at image frame 754 — this event coincides with the first cross-track link. Information from that spatial measurement is propagated along the network topology to other vehicle poses via the shared correlations in the covariance matrix.

3. Thirdly, referring back to Fig. 2-13, note that a temporal (green) link does not exist between consecutive image frames near XY location $(-4, 0)$. A break like this in the temporal image chain prevents concatenation of the relative camera measurements and in a purely vision-only approach could cause algorithms that depend on a connected topology to fail. It is a testament to the robustness of VAN that a disconnected camera topology does not present any significant issue. The key to its success is that navigation allows correlation to be built between temporal poses even though a camera measurement may not exist.

4. Finally, an additional point worth mentioning is that VAN results in a self-consistent estimate of the vehicle's trajectory. Referring to Fig. 2-14, initial processing of the image sequence resulted in a VAN trajectory estimate that did not lie within the 3-sigma confidence bounds predicted by DR. In particular, VAN recovered a crossing trajectory similar to Fig. 2-13 while the DR estimate consisted of two parallel South/North tracklines. Upon further investigation it became clear that the cause of this discrepancy was a significant nonlinear heading bias in the magnetic flux gate compass. We used an independently collected dataset to calculate a compass bias correction and then applied it to our heading data to produce the results shown in Fig. 2-13 where DR and VAN are now in agreement. Essentially, VAN camera-derived measurements had been good enough to compensate for the large heading bias and still recover a consistent vehicle trajectory despite the unmodeled compass error (recall that in a Kalman update the prior will essentially be ignored if the measurements are very certain).

Figure 2-12 Time-evolution of the Stellwagen Bank pose network uncertainty. For each delayed-state entry in $\boldsymbol{\xi}_t$ (i.e., for each vehicle pose $\mathbf{x}_i$), the trace of its 2×2 XY covariance matrix is plotted versus image frame number. The colored dots depict the pose uncertainty at image insertion and the lines show their time evolution for every 5^{th} delayed-state. A couple key events are worth pointing out. First, notice the monotonically increasing uncertainty in XY position between frames 700–753. This period corresponds to when only temporally consecutive image frame measurements could be made. Second, notice the regional smoothing and sharp decrease in uncertainty for correlated state poses at frame number 754. Frame 754 corresponds to the first cross-track spatial measurement made by the camera. Finally, note that the uncertainty in XY pose continues to decrease from frame number 754–763 as more cross-track measurements are made. From frame 764 onward, uncertainty begins to increase again as no more spatial measurements can be made. Shown below the covariance plot is a bar graph of the number of successfully registered image pairs for each frame number. Temporally consecutive camera measurements are shown in green, and the number of spatial cross-track measurements are in red. Notice that the decrease in XY uncertainty in the covariance plot coincides with the first cross-track measurement made by the camera (frame number 754).

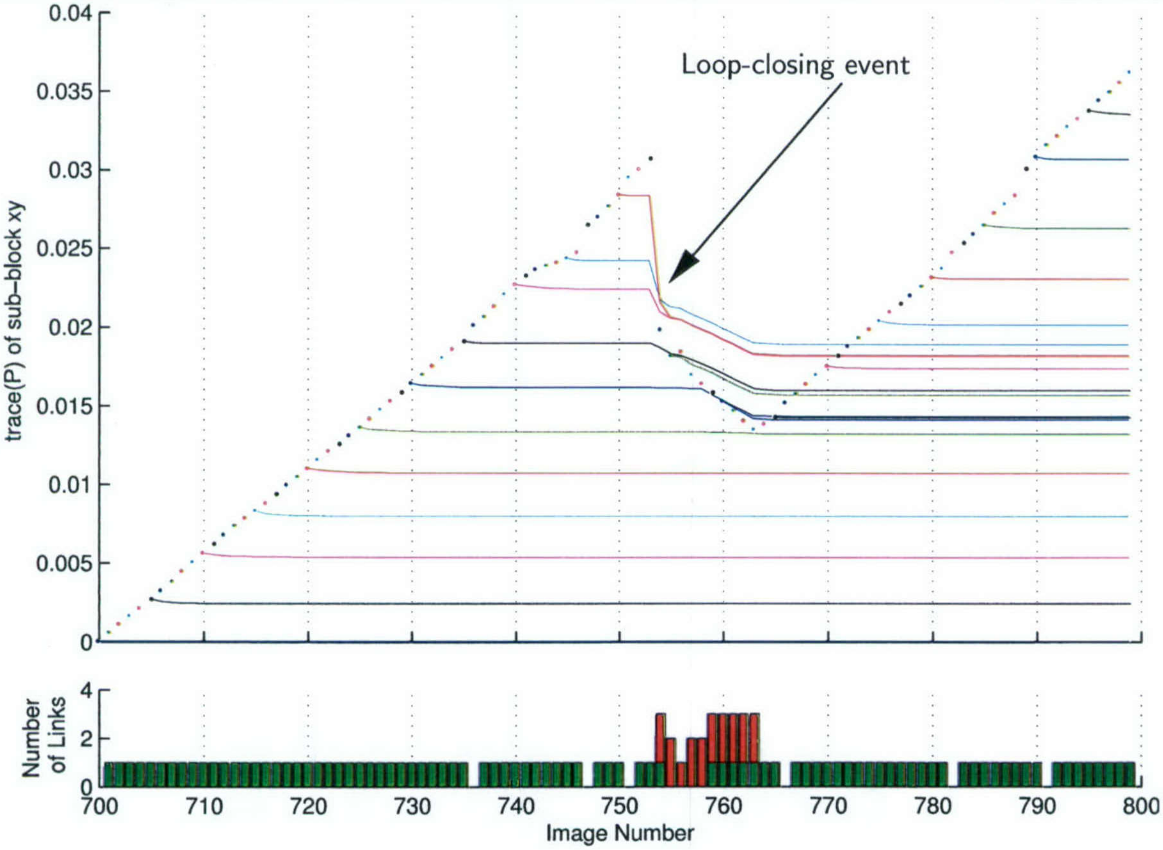
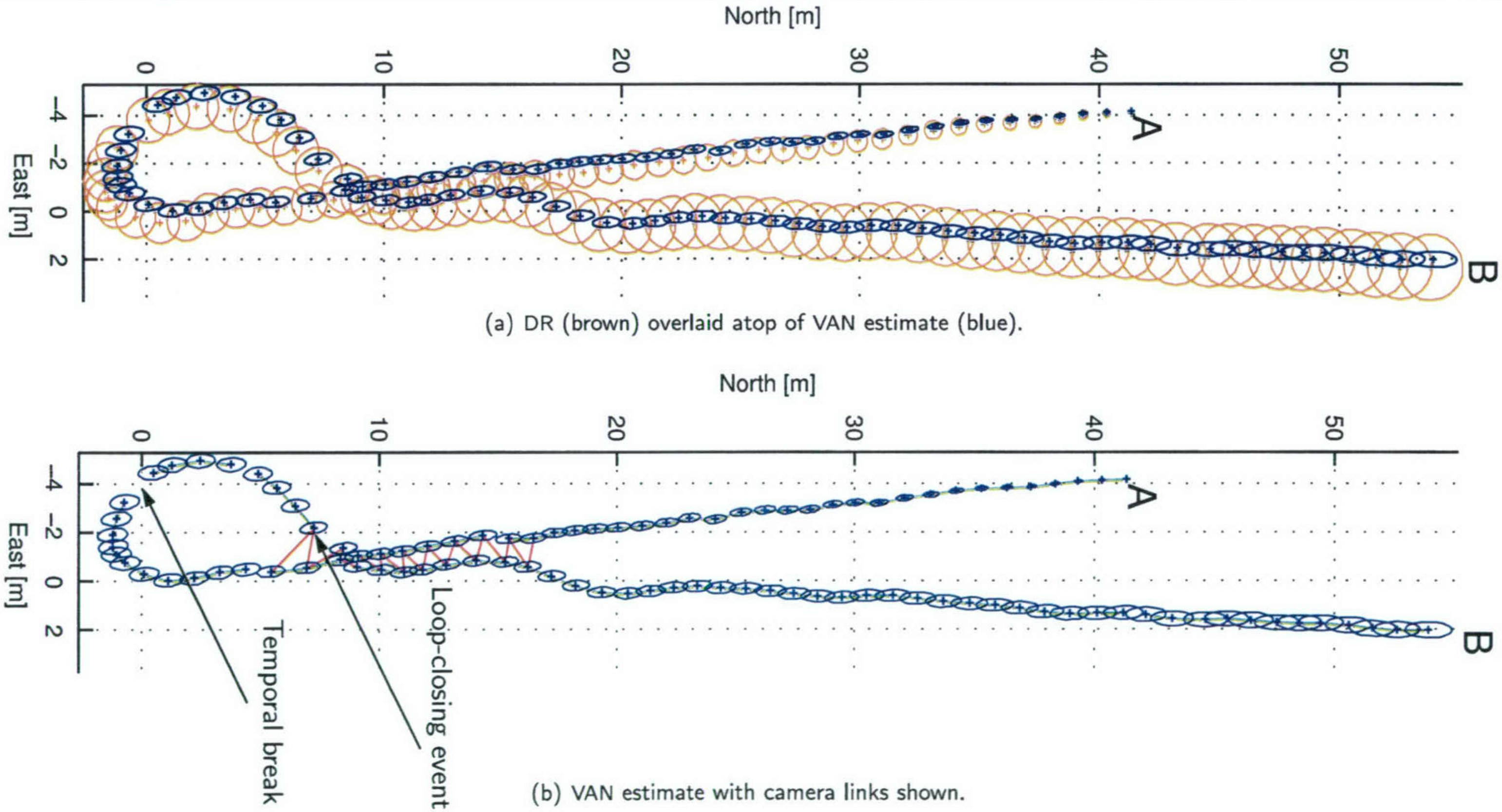


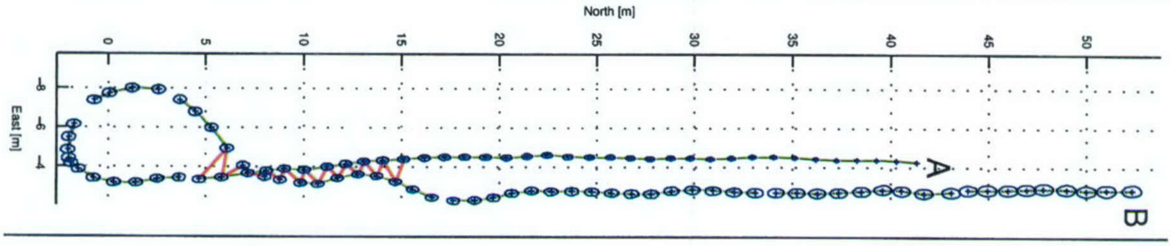
Figure 2-13 Comparison of VAN and DR recovered trajectories; the survey started at A and ended at B. (a) Shown in blue is the XY plot of 100 estimated camera poses with 3-sigma confidence ellipses. For comparison, overlaid in brown is the DR estimated trajectory (also with 3-sigma confidence ellipses). Notice that the DR error monotonically increases, while in contrast, the VAN error is bounded for images in the vicinity of the cross-over point where spatial camera measurements occur. (b) The same 100 estimated camera poses, but with image measurement links superimposed. The green links in the recovered trajectory indicate that a camera-derived measurement was made between temporally consecutive image pairs, while the red links represent that a cross-track spatial measurement was made between the indicated image pairs. In all, 19 cross-track spatial measurements were made. Notice the absence of a temporal camera measurement near $(-4, 0)$. The VAN framework gracefully handles having a disconnected temporal image chain topology since navigation sensors continue to build correlation between camera poses.



(a) DR (brown) overlaid atop of VAN estimate (blue).

(b) VAN estimate with camera links shown.

Figure 2-14 VAN recovers a camera-consistent trajectory estimate despite an unmodeled compass heading bias. The DR trajectory (gray) does not lie within the 3-sigma VAN estimate (blue). This discrepancy comes from an unmodeled compass bias, which when accounted for, produces the results shown in Fig. 2-13. It is a testament to the robustness of VAN that camera-derived relative-pose measurements forced the recovery of a consistent cross-over trajectory despite having an unmodeled heading bias.



2.5.2 Experimental Validation: JHU Test Tank

To validate the error characteristics of VAN as compared to traditional DR, we collaborated with our colleagues at JHU to collect an in-tank ROV dataset with ground truth.[9]

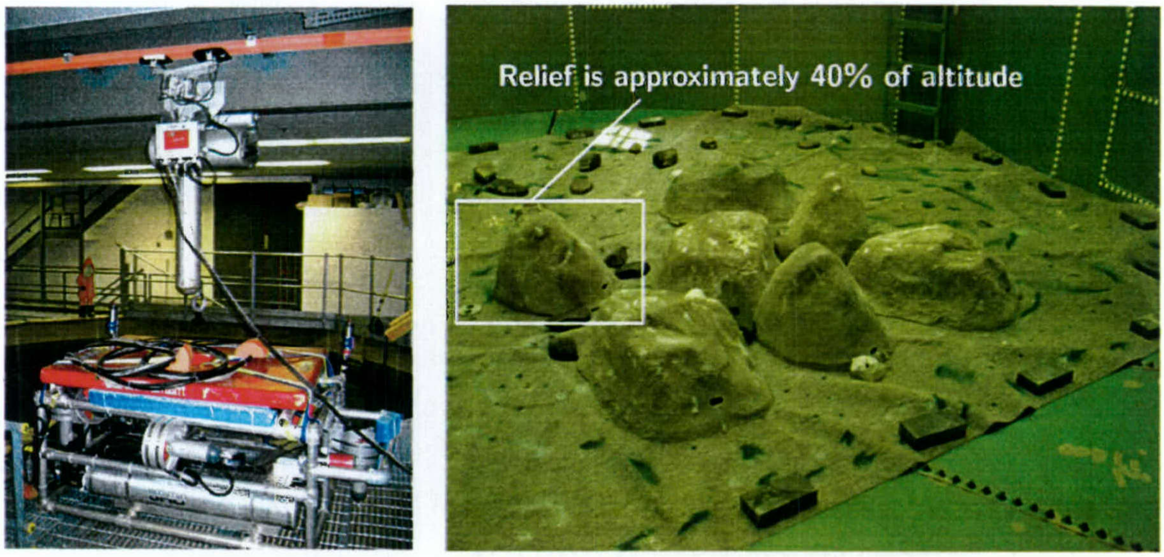
Experimental Setup

The experimental setup consisted of a single downward-looking digital still camera mounted to a moving underwater pose instrumented ROV at the JHU Hydrodynamic Test Facility. Their vehicle is instrumented with a typical suite of oceanographic dead-reckoning navigation sensors capable of measuring heading, attitude, XYZ bottom-referenced Doppler velocities, and a pressure sensor for depth. The vehicle and test facility are also equipped with a high frequency acoustic LBL system, which provides centimeter-level bounded error XY vehicle positions used for validation purposes only. A simulated seafloor environment was created by placing textured carpet, riverbed rocks, and landscaping boulders on the tank floor and was appropriately scaled to match a rugged seafloor environment with considerable 3D scene relief. See Fig. 2-15 for depiction of the experimental setup.

In addition, we also tested an innovative dual-light/camera configuration by placing fore and aft lights on the ROV with the camera mounted in the center as shown in Fig. 2-16. The purpose of this test was to see if we could mitigate the viewpoint variant illumination effects that had prevented us from automatically establishing cross-track correspondences in the Stellwagen Bank dataset. As the results in this section show, the outcome was successful. The dual-light configuration alleviated viewpoint illumination effects by improving the signal-to-nose ratio in shadowed regions so that fully automatic cross-track correspondences were achieved.

[9] We graciously thank our hosts Louis Whitcomb and James Kinsey for their collaboration in collecting the JHU dataset.

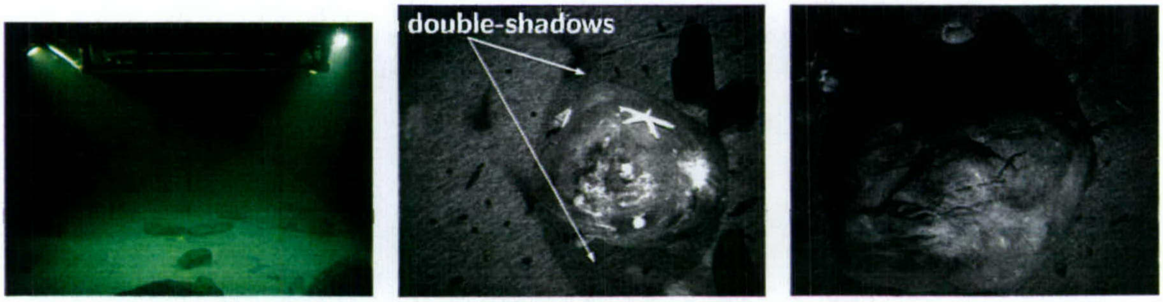
Figure 2-15 The JHU experimental setup. Low-pile carpet, artificial landscaping boulders, and riverbed rock were all placed on the tank floor to create a natural looking seafloor with extensive scene relief for a camera altitude of 1.5 m.



(a) JHU ROV.

(b) Partial view of the artificial scene constructed on the tank floor.

Figure 2-16 The dual-light setup used on the JHU ROV. This experimental dual-light configuration with the camera mounted in the center made fully automatic image registration robust to the effects of viewpoint variant scene illumination. Traditional single-light configurations (c) cast significant shadows and cause objects to look very different from differing vantage points. This makes automatic correspondence establishment very difficult. Meanwhile, the experimental dual-light configuration (b) increases image illumination invariance by casting "double-shadows" that the camera can "see through".

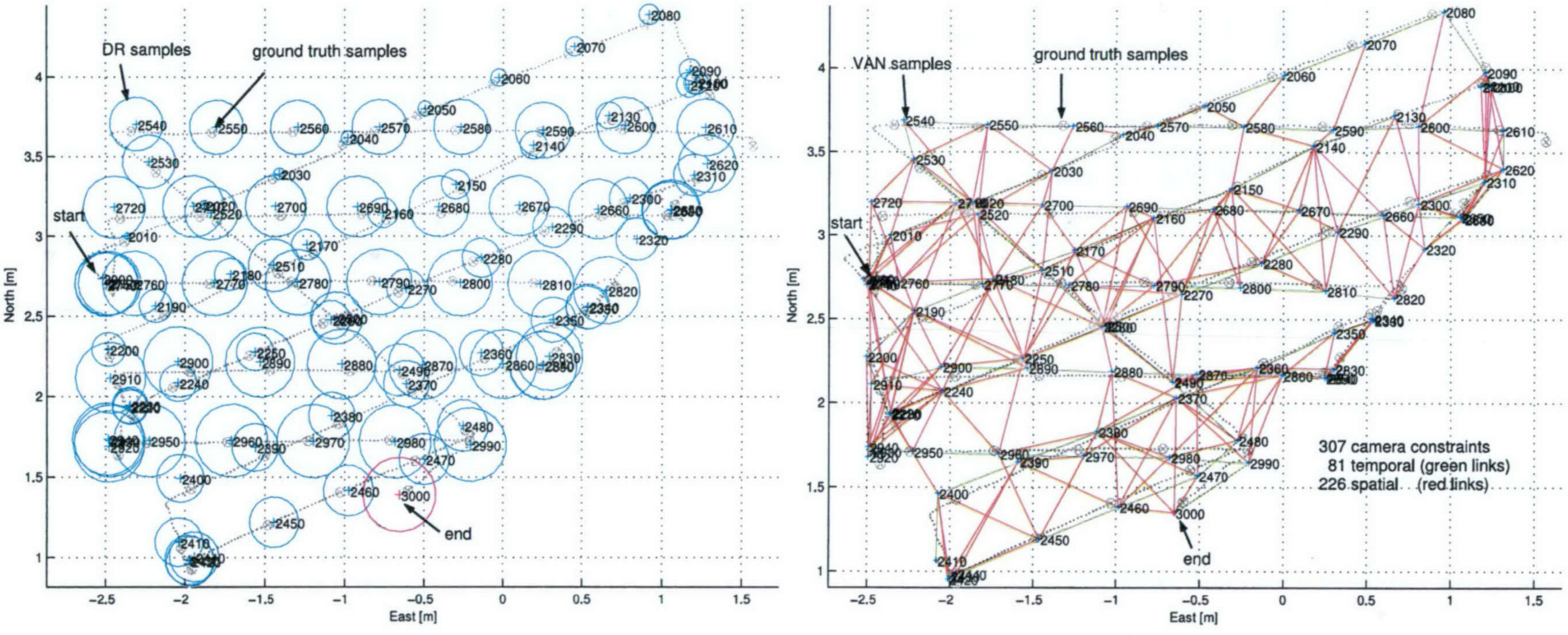


(a) Dual-light configuration.

(b) Dual-light illumination.

(c) Single-light illumination.

Figure 2-17 JHU tank results comparing DR and VAN trajectories to 300 kHz LBL ground-truth for a 101 image sequence. We subsampled the image sequence using only every 10th frame to give roughly 25% temporal overlap (frame numbers start at 2000). The survey consisted of two overlapping grid trajectories, one oriented Northeast/Southwest and the other East/West. The vehicle pose samples and 3-sigma confidence ellipses are shown for all 101 views. The corresponding time samples from the ground truth trajectory are designated by the gray circles. The VAN result is end-to-end fully automatic including link hypothesis (Algorithm 2.1) and correspondence establishment. Notice that XY uncertainty grows monotonically in the DR trajectory estimate while for VAN it is constrained by the camera-constraint topology.



(a) DR trajectory.

(b) VAN trajectory.

Experimental Results

Fig. 2-17 shows the estimated XY trajectory for a 101 image sequence comprised of roughly 25% temporal overlap. For this experiment, the vehicle started near the top-left corner of the plot at (-2.5,2.75) and then drove a course consisting of two grid-based surveys, one oriented SW to NE and the other W to E. Both plots show the spatial XY pose topology, 3-sigma confidence bounds, and network of camera constraints — note that the VAN result is end-to-end fully automatic. Green links correspond to temporally consecutive images that were successfully registered while red links correspond to spatially registered image pairs — in all there are 307 camera constraints (81 temporal / 226 spatial). Notice that the XY uncertainty in the DR estimate grows monotonically with time while in the VAN estimate it is constrained by the camera-link topology. Fig. 2-18 further corroborates this observation and in particular Fig. 2-18(b) shows that VAN exhibits a linear trend in error growth as a function of distance away from the reference node. Note that the spread of points away from this linear fit is due to inhomogeneity in the number of edges per node in the corresponding pose-constraint network. Nonetheless, this raises the interesting engineering question of how one might go about reducing the slope of the linear relationship exhibited in Fig. 2-18(b)? From a camera perspective, design criteria that could help improve this performance are:

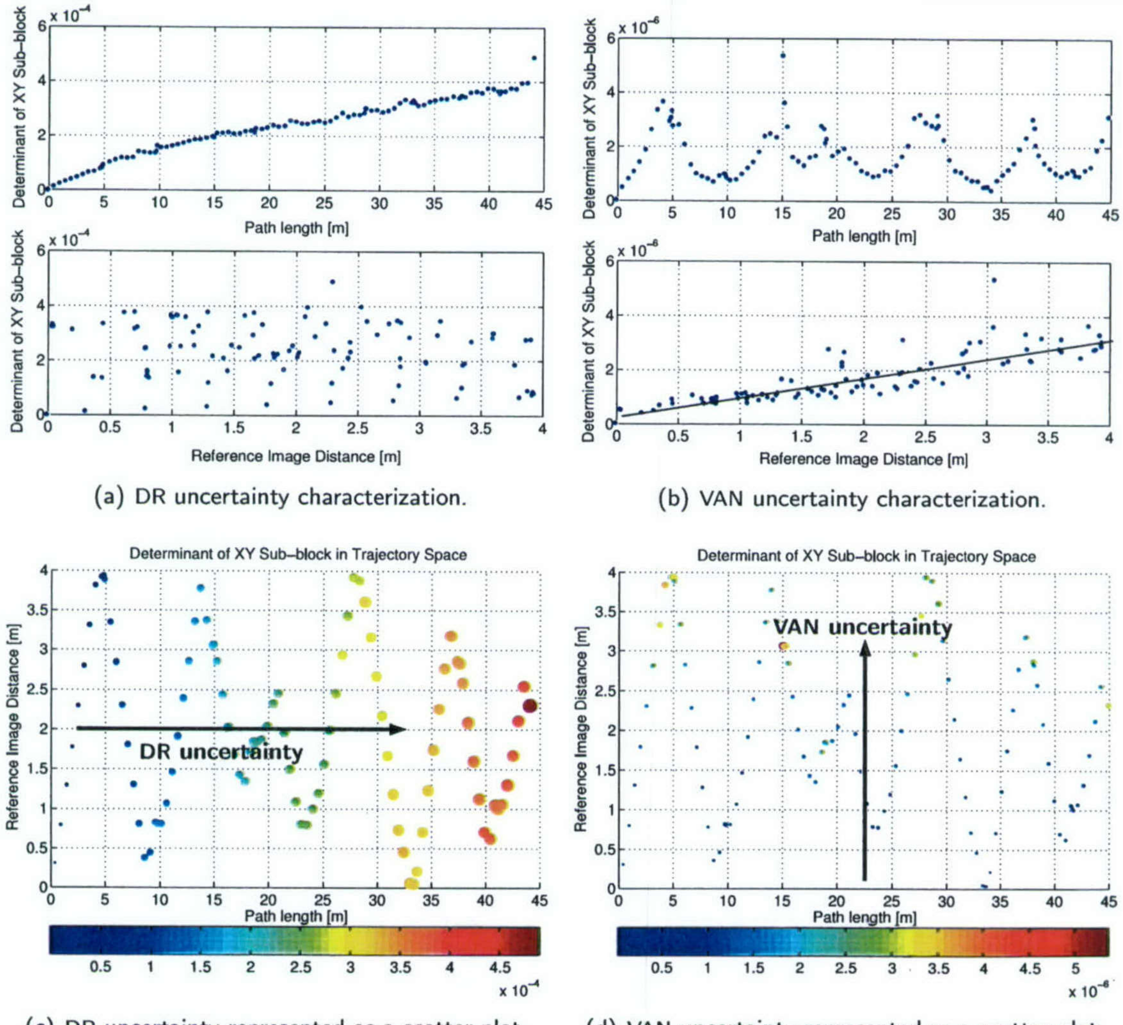
- Higher resolution images. Increased resolution improves both the accuracy and precision with which 2D feature points can be extracted and localized within the viewable image plane. This in turn improves the accuracy and precision of the relative-pose camera measurement.

- A wider FOV. Increasing the camera's FOV improves the pairwise observability of camera motion and, hence, the overall precision of the camera-derived relative-pose measurement. However, increasing the FOV also results in lower image resolution, so a good balance between the two should be found.

- Characterize feature repeatability. Recall that our image registration module employs the common assumption that features are extracted with independent, isotropic, unit variance pixel noise. This noise model does not have any real physical basis, but rather is assumed merely for convenience. Hence, it would be worthwhile to setup a testbed of seafloor imagery for measuring the repeatability of our image feature extractors under different viewing, surface, and lighting conditions. This would provide a more accurate characterization of the feature extraction precision and, thus, a better description for the overall precision of our relative-pose camera measurements.

- Better camera calibration. Our registration framework assumes that we are using a calibrated camera, which implies that the projective mapping from Euclidean ray space to image pixel space is known. Therefore, a poor calibration could introduce a persistent bias into our camera-derived relative-pose measurements and effect the overall consistency of the state estimate. Hence, obtaining a good calibration is important.

Figure 2-18 Uncertainty characteristics of VAN versus DR for the JHU tank dataset shown in Fig. 2-17. Plots (a) and (b) show the determinant of the XY sub-block for each vehicle pose in the view-based map. The determinant is plotted versus both path length and Euclidean distance away from the first image in the view-based map. Notice that the DR uncertainty is clearly a monotonic function of path length whereas VAN uncertainty is related to the distance away from the reference image. Plots (c) and (d) show the same trend but represented in a different way. Here, the XY pose determinant for each view is plotted in a scatter plot versus both reference image distance and path length — the size and color of each disk is proportional to the determinant. In this representation we see that VAN and DR have orthogonal uncertainty characteristics with DR growing per path length (which is a proxy for time) and VAN growing per distance away from the anchor image.



(a) DR uncertainty characterization.

(b) VAN uncertainty characterization.

(c) DR uncertainty represented as a scatter plot.

(d) VAN uncertainty represented as a scatter plot.

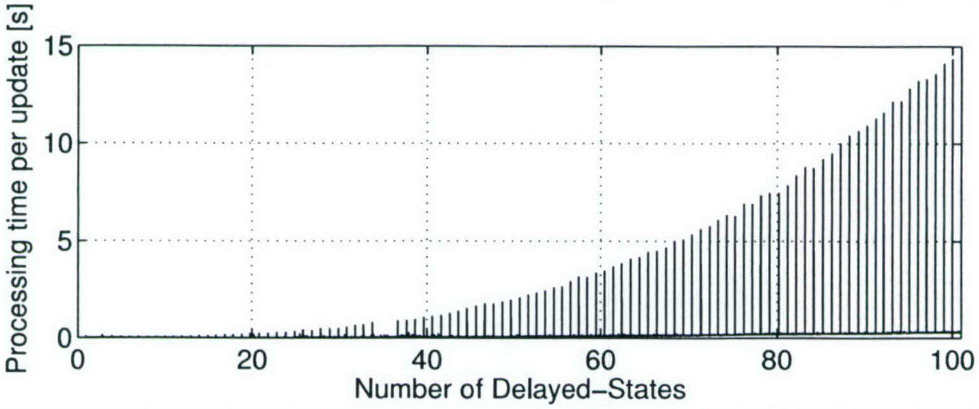
2.6 Chapter Summary

This chapter presented a systems-level framework for visual navigation termed "visually augmented navigation". VAN's systems-level approach leads to a robust solution that exploits the complementary characteristics of a camera and strap-down sensor suite to overcome the peculiarities of low-overlap underwater imagery. Key strengths of the VAN framework were shown to be:

- Self-consistency. Camera measurements forced the VAN trajectory of Fig. 2-14 to "cross-over" despite the presence of an unmodeled compass bias.
- Robustness. Trajectory estimation gracefully handles having a disconnected temporal image chain since navigation builds correlation between camera poses.
- Smoothing. The delayed-state EKF framework means that information from loop-closing events gets distributed throughout the entire map via the joint-correlations.
- Bounded error characteristics. Uncertainty in a DR system grows monotonically time, while in a VAN approach it is a function of network topology. Essentially, VAN allows error to be a function of space and not time — space being distance away from the reference node in a connected topology.

While the above strengths are promising attributes of a standalone precision navigation system, it is well known that a vanilla EKF SLAM implementation is limited to relatively small environments due to the $\mathcal{O}(n^2)$ computational complexity per update to maintain the covariance matrix (Fig. 2-19). In practical terms, this implies that VAN can only maintain a hundred or so images in its view-based map. In Chapters 3 and 4 we address this scalability issue by exploring the re-parameterization of our state estimate in terms of the dual of an extended Kalman filter — an extended information filter. In particular, Chapter 3 discusses feature-based SLAM and the recently derived sparse extended information filter (SEIF) by Thrun et al. [160]. SEIFs relies upon making approximations in the information form to obtain a sparse representation allowing for efficient updates. However, as we show, this approximation leads to unintended consequences regarding filter consistency. Meanwhile, Chapter 4 presents the novel insight that a view-based SLAM framework has *exact* sparsity in the information form. The implication of this is that we can retain VAN's promising navigation attributes while exploiting the sparse representation to achieve $\mathcal{O}(n)$ algorithmic complexity.

Figure 2-19 An EKF's update time grows quadratically with state size. This figure depicts CPU time versus map size for the JHU dataset of Fig. 2-17.



CHAPTER 3

Sparse Extended Information Filters: Insights into Sparsification

RECENTLY, there have been a number of variant simultaneous localization and mapping algorithms that have made substantial progress towards large-area scalability by parameterizing the SLAM posterior within the information (canonical/inverse covariance) form. Of these, probably the most well-known and popular approach is the sparse extended information filter (SEIF) by Thrun et al. [160]. While SEIFs have been successfully implemented with a variety of challenging real-world data sets and have lead to new insights into scalable SLAM, open research questions remain regarding the approximate sparsification procedure and its effect on map error and consistency.

In this chapter, we examine the constant-time SEIF sparsification procedure as it pertains to feature-based SLAM and offer new insight into issues of consistency. In particular, we show that exaggerated map inconsistency occurs within the *global* reference frame where estimation is performed, but that empirical testing shows that *relative* map relationships are preserved. We then present a slightly modified version of their sparsification procedure, which is shown to preserve sparsity while also generating both local and global map estimates comparable to those obtained by the non-sparsified SLAM filter (i.e., full-covariance EKF). This modified approximation, however, is no longer constant-time. We demonstrate our findings by benchmark comparison of the modified and original SEIF sparsification rules using a linear Gaussian SLAM simulation and a real-world experiment for a nonlinear dataset. From this chapter we conclude that *approximate* sparsification is a necessary requirement for feature-based SLAM to be efficient in the information form — however, this approximation is non-trivial.

3.1 Introduction

Since its inception with the fundamental work of Smith, Self, and Cheeseman [154] and Moutarlier and Chatila [110], roboticists have been trying to address scalability issues associated with an extended Kalman filter based approach to SLAM. While this approach is often considered the "standard" [30] and is attractive in its simplicity (because it only requires tracking first and second moments of the joint landmark-robot distribution), a well known fact is that EKF SLAM inference requires quadratic complexity in the number of landmarks *per update* to maintain the joint-posterior correlations. As a consequence, the direct application of EKF SLAM is limited to relatively small environments (e.g., less than than 100 landmarks).

3.1.1 A Survey of Scalable Feature-Based SLAM Algorithms

Submaps

Over the years a number of different approaches have been put forth to try and curb this quadratic cost with map size. One of the more conceptually straightforward approaches has been the idea of dividing the world into local submaps [11, 85, 86]. Submaps are based upon decomposing the environment into manageable "local pieces" thereby bounding each map's computational growth by limiting its size (both in a physical sense and in terms of the number of local features it contains). While these approaches have been demonstrated to be computationally efficient for mapping large cyclic environments (for example the *Atlas* framework by Bosse et al. [11], and the CTS algorithm by Leonard and Newman [86]), they sacrifice convergence rates by not explicitly enforcing measurement constraints in full across the network of submaps [86]. In addition, the definition of a submap tends to be rather ad-hoc, which leads to a poor representation for the border features [160].

Postponement

Another closely related idea to submaps is the concept of postponement [27] and its variants [173]. Postponement is built around the standard EKF SLAM approach, but employs a clever "bookkeeping" technique to the update equations whereby the robot is able to efficiently work in a "local" map context without the burden of maintaining the "global" map. It is proven to yield mathematically equivalent results to the full EKF implementation while simultaneously relieving the computational load when working locally; as long as the robot observes elements within the local environment, measurements can be efficiently incorporated with bounded complexity. Upon moving to a new area, however, the robot must first fuse the local-map into the global representation at the standard $\mathcal{O}(n^2)$ cost (where n is the total number of *global* landmarks) — hence the name "postponement".

FastSLAM

In addition to the EKF approaches, FastSLAM [108] has appeared in the recent literature as a large-scale SLAM algorithm based upon an entirely different representation of the joint landmark-robot posterior. FastSLAM employs Rao-Blackwellized particle filtering [32] to decouple the joint landmark-robot distribution into n independent landmark estimation problems by using a particle filter to sample over the robot trajectory. This landmark factorization is exact based upon each particle representing a realization of the robot's path and leads to an $\mathcal{O}(m \log n)$ update complexity where m is the number of particles used to represent the trajectory. The algorithm has the benefit of intrinsically handling multiple data association hypotheses (i.e., the problem of establishing correspondences to measurements [118]) and its scalability has been demonstrated in simulation by building maps with over 50,000 features [108]. However, as a significant drawback to its general applicability, the theoretical relationship between the size of the mapped area and the number of required particles is poorly understood [160] (and conjectured to even be exponential in environmental size [161]).

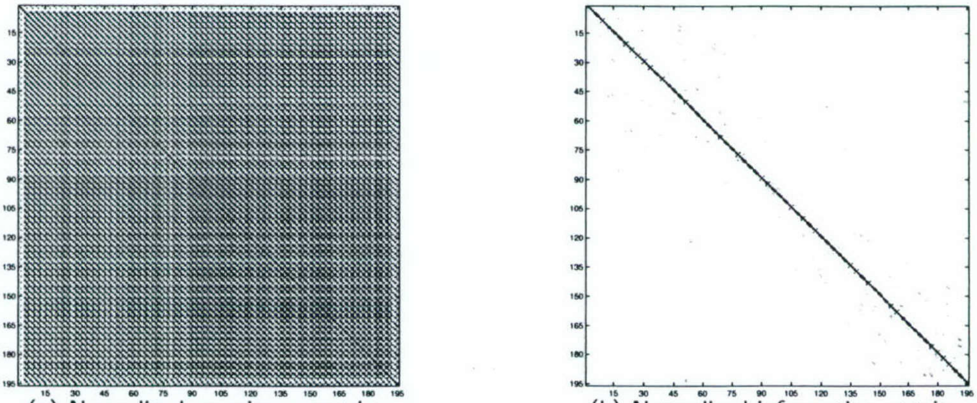
Covariance Intersection

Lastly, the technique of covariance intersection (CI) [75] represents yet another approach to large scale map making distinctly different from the EKF framework and additionally has the benefit of constant-time fusion with linear storage. It achieves low computational complexity by essentially ignoring the joint landmark covariances and adopting a conservative fusion strategy that respects all possible correlations [23]. The empirical result has been that while CI can handle a million landmark map [75], its fusion strategy is so conservative that practical convergence rates cannot be achieved.

3.1.2 The Information Form and a New Class of Algorithms

Recently, a new class of scalable SLAM algorithms have been proposed by Thrun et al. [160], Paskin [120], and Frese [49, 51], and are all based upon the canonical-form. This representation has the nice interpretation as a Gaussian graphical model [120, 135]. As Thrun et al. [160] empirically first showed, and Frese later analytically proved [50], the inverse covariance matrix (i.e., information matrix) of feature-based SLAM exhibits a "natural" sparseness whereby many of the off-diagonal elements (i.e., graphical constraints) are relatively "weak" (see Fig. 3-1). This insight has spawned the development of scalable SLAM algorithms founded upon pruning these weak constraints and exploiting the resulting sparse representation [49, 120, 160].

Figure 3-1 Feature-based SLAM exhibits a "natural" sparsity in the information form. A comparison of the structure of the covariance and information matrices as is typically seen in feature-based SLAM implementations; darker shades represent larger magnitudes. (a) The correlation matrix is dense and requires quadratic storage. (b) Normalizing the information matrix in the same manner as the correlation matrix yields an "almost sparse" representation where a majority of the elements are orders of magnitude smaller than the dominant entries.



(a) Normalized covariance matrix.

(b) Normalized information matrix.

For example, both Paskin (Thin-Junction-Tree Filters) [120] and Frese (Treemap Filters) [49] employ tree-based approximations to sparsify the canonical-form and have developed very efficient inference algorithms for this representation. One drawback to their techniques, though, is that their tree-based representations cannot explicitly model cyclic environments nor has data association been addressed. Alternatively, the sparse extended information filter proposed by Thrun et al. [160], probably the most well known SLAM information formulation, is based upon representing the SLAM posterior through the dual of the extended Kalman filter. SEIFs maintain a sparse information matrix, an approach which has been demonstrated to be both efficient and scalable, allows for explicit representation of cyclic environments, and addresses data association [92]. The delicate issue, however, is *how* to perform the necessary sparsification step required to keep the information matrix sparse.

In this chapter, we explore in depth the approximation employed by SEIFs to enforce sparseness. We show that a particular assumption in its derivation leads to inconsistency of the global map error covariance estimates, however, empirical testing indicates that local map relations and relative uncertainties are preserved. In addition, we present a slightly modified derivation that yields an alternative sparsification rule, which is shown to produce both global and local map estimates comparable to the full-covariance EKF while also maintaining the same level of sparsity as SEIFs. Its drawback, though, is that sparsification is no longer constant-time. We demonstrate our insights by concluding with a benchmark comparison for a linear Gaussian SLAM simulation and in addition present results for a nonlinear experimental dataset.

3.2 Background

3.2.1 The Gaussian Information Form

The information form is often called the canonical or natural representation of the Gaussian distribution. This notion of a "natural" representation stems from expanding the quadratic in the exponential of the Gaussian distribution as

$$\begin{aligned}
p(\boldsymbol{\xi}_t) &= \mathcal{N}\big(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t\big) \\
&= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\Big\{-\tfrac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)^\top \Sigma_t^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)\Big\} \\
&= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\Big\{-\tfrac{1}{2}\big(\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t - 2\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t\big)\Big\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t}}{\sqrt{|2\pi\Sigma_t|}} \exp\Big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t\Big\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\eta}_t^\top \Lambda_t^{-1}\boldsymbol{\eta}_t}}{\sqrt{|2\pi\Lambda_t^{-1}|}} \exp\Big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top \Lambda_t\boldsymbol{\xi}_t + \boldsymbol{\eta}_t^\top \boldsymbol{\xi}_t\Big\} \\
&= \mathcal{N}^{-1}\big(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t\big)
\end{aligned}$$

where $\Lambda_t = \Sigma_t^{-1}$ and $\boldsymbol{\eta}_t = \Lambda_t\boldsymbol{\mu}_t$. The result is that rather than parameterizing the normal distribution in terms of its mean and covariance, $\mathcal{N}\big(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t\big)$, it is instead parametrized in terms of its information vector and information matrix, $\mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)$ [7].

3.2.2 Marginalization and Conditioning

The covariance and information representations lead to very different computational characteristics with respect to the fundamental probabilistic operations of marginalization and conditioning. This is important because these two operations appear at the core of any SLAM algorithm, for example motion prediction and measurement updates. Table 3.1 summarizes these operations for a Gaussian distribution where we see that the covariance and information representations exhibit a dual relationship with respect to marginalization and conditioning. For example, marginalization is easy in the covariance form since it corresponds to extracting the appropriate sub-block from the covariance matrix while in the information form it is hard because it involves calculating the Schur complement over the variables we wish to keep. Notice that the opposite relation holds true for conditioning, which is easy in the information form and hard in the covariance form.

Table 3.1 Summary of the marginalization and conditioning operations for the Gaussian distribution as expressed in both the covariance and information form.[a]

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{bmatrix}, \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}\right) = \mathcal{N}^{-1}\left(\begin{bmatrix} \boldsymbol{\eta}_\alpha \\ \boldsymbol{\eta}_\beta \end{bmatrix}, \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}\right)$$

	Marginalization $p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\beta}$	**Conditioning** $p(\boldsymbol{\alpha} \mid \boldsymbol{\beta}) = p(\boldsymbol{\alpha}, \boldsymbol{\beta}) / p(\boldsymbol{\beta})$
Cov. Form	$\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$	$\boldsymbol{\mu}' = \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)$ $\Sigma' = \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}$
Info. Form	$\boldsymbol{\eta} = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta$ $\Lambda = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$	$\boldsymbol{\eta}' = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}$ $\Lambda' = \Lambda_{\alpha\alpha}$

[a] A derivation of these operations can be found in Appendix §B.1.

3.2.3 Controlling Feature-Based SLAM Sparsity

Most SLAM approaches are feature-based, which assumes that the robot can extract an abstract representation of features in the environment from its sensor data and then use re-observation of these features for localization [154]. In this approach, a landmark map is explicitly built and maintained. The process of concurrently performing localization *and* map building are inherently coupled, therefore, the robot must then represent a joint-distribution over landmarks and current pose, i.e.,

$$p(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t) = \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t) = \mathcal{N}^{-1}(\boldsymbol{\eta}_t, \Lambda_t) \tag{3.1}$$

where $\boldsymbol{\xi}_t = [\mathbf{x}_t^\top, \mathbf{M}^\top]^\top$ represents the current robot state and landmark map respectively, $\mathbf{z}^t$ the set of measurements up to time t, and $\mathbf{u}^t$ the set of control inputs. In (3.1) we have explicitly modeled the distribution as being jointly-Gaussian based upon additive white noise models and first-order linearizations of our process and observation models as described in [154, 160]. The key behind scalable SLAM algorithms in the canonical-form is founded upon the insight that the information matrix Λ_t *naturally* tends to exhibit strong and weak constraints as shown by Fig. 3-1.

Filtering Causes Fill-in

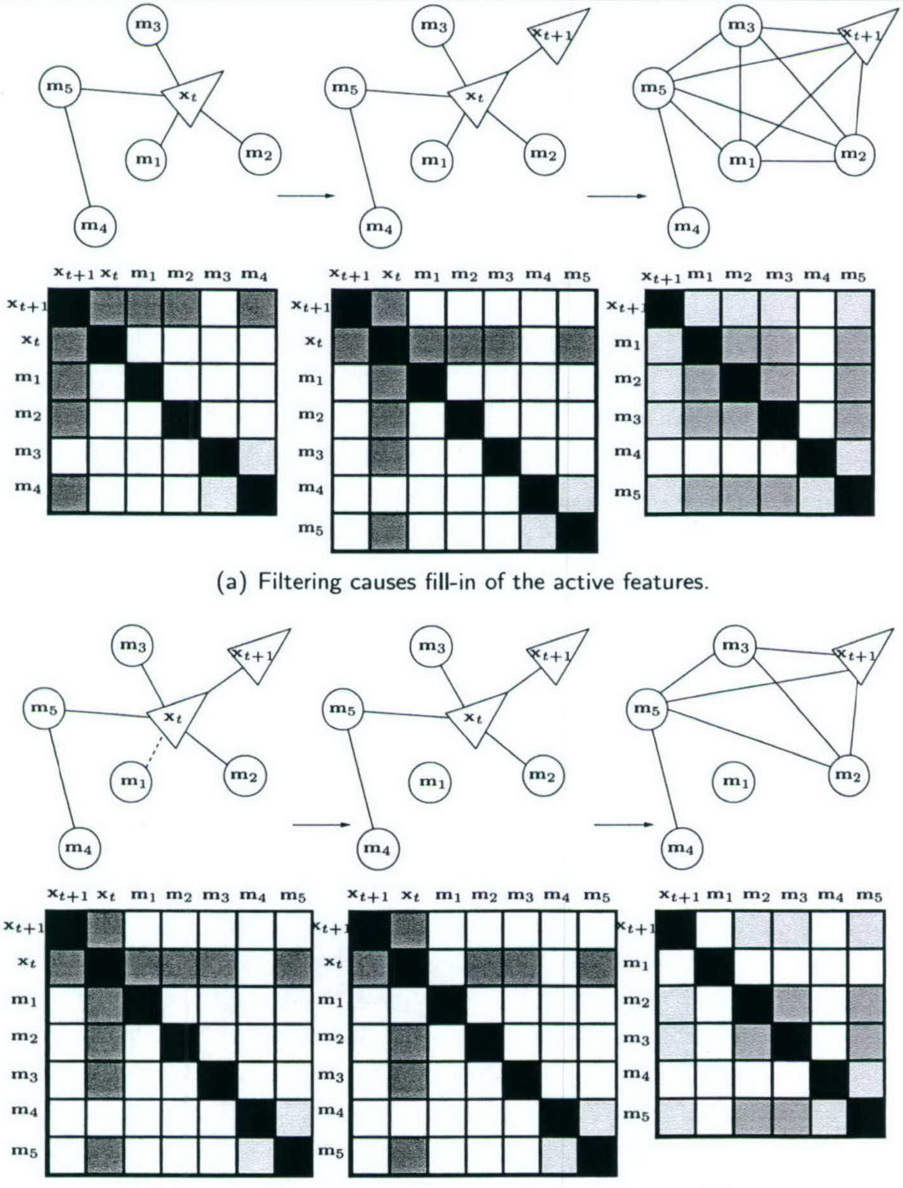
What Thrun et al. [160] insightfully observed is that the time-projection step (i.e., motion prediction) is the cause for creating these weak constraints. Furthermore, by bounding the number of nonzero off-diagonal elements linking the robot to landmarks, many of these weak constraints can be eliminated. Their concept is to partition the landmark map $\mathbf{M}$ into a set of active features $\mathbf{m}^+$ (i.e., those with a nonzero off-diagonal element linking them to the robot $\mathbf{x}_t$) and a set of passive features $\mathbf{m}^-$ (i.e., those with no link to $\mathbf{x}_t$).[1] They showed that by enforcing an upper bound on the number of active features $\mathbf{m}^+$, the fill-in of the information matrix can be controlled.

To see this, consider the diagram shown in Fig. 3-2(a). We begin with the schematic shown to the upper left, which represents the robot, $\mathbf{x}_t$, at time t connected to four active landmark map elements, $\mathbf{m}^+ = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_5\}$. Also shown is a single passive map element, $\mathbf{m}^- = \{\mathbf{m}_4\}$, who is not linked to the robot, but only to another landmark, $\mathbf{m}_5$. For now we will ignore how this connection came to be and just take it for granted that it exists. Conceptually, what Fig. 3-2(a) represents is a graphical representation of the conditional independencies in the distribution $p(\mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^t)$ and is known in the literature as a Markov random field (MRF) or Markov network [122]. Therefore, in these terms we see that the robot to active landmark connections are the direct result of perceptual sensing and that the lack of inter-landmark constraints should be correctly interpreted to mean that each of these landmarks is conditionally independent *given* the robot pose as described in [108, 111]. The intuition behind this comes from viewing the noise of each sensor reading as being independent, and therefore, determining each of these landmark positions is an independent estimation problem given the *known* location of the sensor.

Shown directly below each Markov network is an illustration of the associated information matrix. Here we see that the nonzero off-diagonal elements encode the robot/landmark constraints (i.e, edges) while the zeros in the information matrix encode *missing* edges [121]. Moving on to the middle diagram of Fig. 3-2(a), we see that it represents the intermediate distribution $p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^t)$, which corresponds to a time-propagation of $p(\mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^t)$ where we have augmented our state vector to include the term $\mathbf{x}_{t+1}$ representing the new robot pose at time $t+1$. Because the robot state evolves according to a Markov process, the new robot state $\mathbf{x}_{t+1}$ is only linked to the previous robot state $\mathbf{x}_t$. Usually, in a feature-based SLAM approach only the current robot pose is estimated and not the complete trajectory. Therefore, we always marginalize out the previous robot pose $\mathbf{x}_t$ during our time-projection step to yield the distribution over current pose and map $p(\mathbf{x}_{t+1}, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^t) = \int p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^t) d\mathbf{x}_t$. Recalling the formula for marginalization

[1] The total landmark map is given by $\mathbf{M} = \{\mathbf{m}^+, \mathbf{m}^-\}$.

Figure 3-2 A graphical explanation of SEIFs methodology for controlling sparsity in the information matrix. (a) A sequence of illustrations depicting the evolution of the Markov network and corresponding information matrix resulting from time projection when viewed as a two-step process of state augmentation followed by marginalization. Gray shades imply magnitude with white being exactly zero. From left to right we have: (1) the robot $\mathbf{x}_t$ connected to four active features, $\mathbf{m}_{1:3}$ and $\mathbf{m}_5$; (2) state augmentation of the time-propagated robot pose $\mathbf{x}_{t+1}$; (3) marginalized distribution where the old pose, $\mathbf{x}_t$, has been eliminated. (b) A sequence of illustrations highlighting the concept behind sparsification. If feature $\mathbf{m}_1$ can first be made passive by eliminating its link to the old pose, $\mathbf{x}_t$, then marginalization over $\mathbf{x}_t$ will not link it to any of the other active features. This implies that we can control fill-in of the information matrix by bounding the number of currently active features. Note that for both cases the passive feature $\mathbf{m}_5$ remains disconnected.



(a) Filtering causes fill-in of the active features.

(b) Bounding the number of active features controls fill-in.

applied to a Gaussian in the information form (see Table 3.1), we note that it is the the outer product of $\Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\alpha\beta}^{\top}$ that causes the information matrix to fill-in and become dense as shown in the illustration to the far right of Fig. 3-2(a).[2]

Controlling Active Feature Fill-in

Intuitively, the landmarks $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$, and $\mathbf{m}_5$ used to be indirectly connected via a direct relationship with $\mathbf{x}_t$, but now must represent their indirect relationship directly by creating new links between each other. Therefore, the penalty for a filtered feature-based SLAM representation is that the Markov network becomes fully connected and the associated information matrix becomes fully *dense* (though as previously mentioned, the authors of [160] make the empirical observation that many of the off-diagonal elements are relatively small). Insightfully, this understanding of fill-in suggests that we can *control* the active feature density in the information matrix by *bounding* the number of links connected to $\mathbf{x}_t$ *before* marginalization occurs as illustrated by Fig. 3-2(b). This key insight motivates the concept behind SEIFs sparsification methodology, which is the process of how we remove links to satisfy our active feature bound and maintain a sparse graph representation. But before moving on to discuss *how* SEIFs actually enforce the upper bound on the number of active features, it is useful to first elucidate the conditional independence relationship implied by active and passive features.

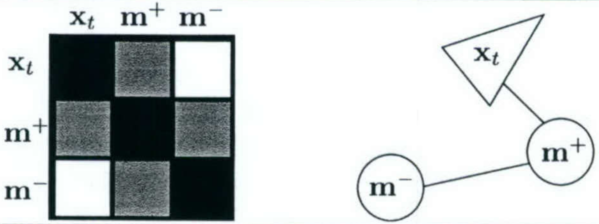
3.2.4 Robot Conditional Independence from Passive Features

A very useful property of the canonical-form is that the information matrix has the direct interpretation as a Gaussian Markov random field (GMRF) [168] where random variables are nodes, non-zero off-diagonal elements are edges/constraints, and zero-valued off-diagonal elements are *missing* edges implying available conditional independencies [135]. Applying this property to SEIF's active/passive partitioning of landmarks we see that Fig. 3-3 correctly illustrates the corresponding block information matrix and associated Markov network for the SEIF posterior. In particular, Fig. 3-3 clearly shows that there is a missing edge between the robot, $\mathbf{x}_t$, and the passive features, $\mathbf{m}^-$, implying that the two are conditionally independent *given* the active features, $\mathbf{m}^+$.

Mathematically, we can also easily prove this relationship by recalling that independence implies that the Gaussian conditional posterior $p(\mathbf{x}_t, \mathbf{m}^- \mid \mathbf{m}^+, \mathbf{z}^t, \mathbf{u}^t)$ must have a block-diagonal covariance matrix (i.e., for a Gaussian random variable, uncorrelated is equivalent to independent). In the information form, conditioning on the active features, $\mathbf{m}^+$, corresponds to simply extracting the $\{\mathbf{x}_t, \mathbf{m}^-\}$ sub-block from the information matrix, Λ_t (see Table 3.1). Referring again to Fig. 3-3, we note that this sub-block results in a block-diagonal conditional information matrix over $\mathbf{x}_t$ and $\mathbf{m}^-$ whose inverse is a block-diagonal covariance matrix. Hence, conditional independence is proved. As we show next, we can exploit this conditional independence relationship to derive a sparsification rule that allows us to bound the number of active features.

[2] Here $\boldsymbol{\alpha} = \{\mathbf{x}_{t+1}, \mathbf{M}\}$ and $\boldsymbol{\beta} = \mathbf{x}_t$

Figure 3-3 An illustration of SEIF's concept of active and passive features and their relation to the robot. (left) A schematic of the block 3 × 3 SEIF information matrix. Dark squares correspond to nonzero block-elements while white squares corresponds to exactly zero block elements. (right) The SEIF information matrix expressed as a Markov network. The missing edge between $\mathbf{x}_t$ and $\mathbf{m}^-$ implies available conditional independence given $\mathbf{m}^+$.



3.3 Sparsification

In feature-based SLAM, landmarks become active through observation by causing them to become linked to the robot through a shared off-diagonal constraint. Furthermore, this constraint will decay over time if the landmark is not re-observed [50], but will never become exactly zero (i.e., passive) unless it is "sparsified". Sparsification refers to the pruning operation whereby the weak robot-landmark constraints are removed causing features to be made passive. It is a useful *approximation* that allows sparsity to be enforced in the information matrix by bounding the number of active features as described in §3.2.3.

3.3.1 SEIF Sparsification Rule

Sparsification is required whenever the active feature threshold is exceeded through landmark observation. SEIF's strategy for sparsification is based upon partitioning the landmark map, $\mathbf{M}$, into a union of three disjoint sets, $\mathbf{M} = \{\mathbf{m}^0 \cup \mathbf{m}^+ \cup \mathbf{m}^-\}$, where in a slight abuse of our previous notation, $\mathbf{m}^-$ are the currently passive features which will *remain* passive after sparsifying, $\mathbf{m}^+$ are the currently active features which will *remain* active after sparsifying, and $\mathbf{m}^0$ are the currently active features which will *become* passive after sparsifying.

We begin our derivation of the SEIF sparsification approximation by factorizing the SLAM posterior over the robot and map as

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) && (3.2a) \\ &= p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^- = \boldsymbol{\alpha}) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-). && (3.2b) \end{aligned}$$

For notational convenience we have omitted explicitly writing out the conditioning on $\mathbf{z}^t$ and $\mathbf{u}^t$. The above factorization invokes the available conditional independence between the robot and passive features discussed in §3.2.4 to arbitrarily assign a value to the passive features in the conditional of (3.2b) (i.e., $\mathbf{m}^- = \boldsymbol{\alpha}$) *without* influencing the conditional robot posterior. In other words, $p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \equiv p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+)$. Note that in the original derivation proposed by [160], $\boldsymbol{\alpha}$ is simply set to zero while here we leave it as a free parameter for the purposes of exposition.

The SEIF sparsification approximation is derived from (3.2b) by imposing that $\mathbf{m}^0$ be passive via dropping it from the robot posterior as

$$\begin{aligned}
&\tilde{p}_{\mathrm{SEIFs}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \\
&\quad = p(\mathbf{x}_t | \mathbf{m}^+, \mathbf{m}^- = \boldsymbol{\alpha}) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) && \text{(3.3a)} \\
&\quad = \frac{p_B(\mathbf{x}_t, \mathbf{m}^+ \mid \mathbf{m}^- = \boldsymbol{\alpha})}{p_C(\mathbf{m}^+ \mid \mathbf{m}^- = \boldsymbol{\alpha})} p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-). && \text{(3.3b)}
\end{aligned}$$

Here, (3.3b) merely expresses the conditional of (3.3a) as a ratio and the subscripts p_B, p_C, p_D are used for notational convenience to reference the different pdfs involved in the calculation. While the factorization of (3.2b) is theoretically *exact* due to the conditional independence between $\mathbf{x}_t$ and $\mathbf{m}^-$ given the active features, (3.3) is in *error* because $\mathbf{x}_t$ is no longer conditionally independent of $\mathbf{m}^-$ given only a partial set of the active features (i.e., the set of all active features is $\mathbf{m}^0 \cup \mathbf{m}^+$). This implies that the particular value of $\boldsymbol{\alpha}$ chosen modifies the posterior approximation.

Equations (3.4) and (3.5) summarize the SEIF sparsified posterior as expressed in both covariance and information form respectively — a complete derivation of the resulting posterior can be found in Appendix §B.2. For ease of comparison we use the same notation as [160] where S denotes a projection matrix over the state space $\boldsymbol{\xi}_t$ (e.g., $\mathbf{x}_t = \mathrm{S}_{x_t}^\top \boldsymbol{\xi}_t$ extracts the robot pose). Note that the mean update in (3.4) clearly shows that the original mean vector $\boldsymbol{\mu}_t$ is modified during the sparsification step for values of $\boldsymbol{\alpha} \neq \mathrm{S}_{m^-}^\top \boldsymbol{\mu}_t$, which indicates $\boldsymbol{\alpha}$'s influence on the term $p(\mathbf{x}_t | \mathbf{m}^+, \mathbf{m}^- = \boldsymbol{\alpha})$ used in the approximation of (3.3).[3]

Covariance Form

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_t + \tilde{\Sigma}_t \big(\mathrm{S}_{x_t,m^+} \Sigma_B^{-1} \mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+} \Sigma_C^{-1} \mathrm{S}_{m^+}^\top \big) \boldsymbol{\mu}_\alpha \\
\tilde{\Sigma}_t &= \big(\mathrm{S}_{x_t,m^+} \Sigma_B^{-1} \mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+} \Sigma_C^{-1} \mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-} \Sigma_D^{-1} \mathrm{S}_{m^0,m^+,m^-}^\top \big)^{-1}
\end{aligned} \tag{3.4}$$

where

$$\begin{aligned}
\boldsymbol{\mu}_B &= \mathrm{S}_{x,m^+}^\top (\boldsymbol{\mu}_t + \boldsymbol{\mu}_\alpha) \\
\Sigma_B &= \mathrm{S}_{x,m^+}^\top \Sigma_t \mathrm{S}_{x,m^+} - \mathrm{S}_{x,m^+}^\top \Sigma_t \mathrm{S}_{m^-} \big(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-} \big)^{-1} \mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{x,m^+}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\mu}_C &= \mathrm{S}_{m^+}^\top (\boldsymbol{\mu}_t + \boldsymbol{\mu}_\alpha) \\
\Sigma_C &= \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^-} \big(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-} \big)^{-1} \mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^+}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\mu}_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \boldsymbol{\mu}_t \\
\Sigma_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \Sigma_t \mathrm{S}_{m^0,m^+,m^-}
\end{aligned}$$

with $\boldsymbol{\mu}_\alpha = \Sigma_t \mathrm{S}_{m^-} \big(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-} \big)^{-1} \big(\boldsymbol{\alpha} - \mathrm{S}_{m^-}^\top \boldsymbol{\mu}_t \big)$.

[3] The expression for the sparsified information vector as presented in [160] corresponds to setting $\boldsymbol{\alpha} = \mathrm{S}_{m^-}^\top \boldsymbol{\mu}_t$, (i.e., the mean of the passive features) and not $\boldsymbol{\alpha} = \mathbf{0}$ as stated in their paper.

Information Form

$$\begin{aligned}\tilde{\boldsymbol{\eta}}_t &= \mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_B - \mathrm{S}_{m^+}\boldsymbol{\eta}_C + \mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D \\ \tilde{\Lambda}_t &= \mathrm{S}_{x_t,m^+}\Lambda_B\mathrm{S}^\top_{x_t,m^+} - \mathrm{S}_{m^+}\Lambda_C\mathrm{S}^\top_{m^+} + \mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}^\top_{m^0,m^+,m^-}\end{aligned} \tag{3.5}$$

where

$$\begin{aligned}\boldsymbol{\eta}_B &= \mathrm{S}^\top_{x_t,m^+}\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) - \mathrm{S}^\top_{x_t,m^+}\Lambda_t\mathrm{S}_{m^0}\left(\mathrm{S}^\top_{m^0}\Lambda_t\mathrm{S}_{m^0}\right)^{-1}\mathrm{S}^\top_{m^0}\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) \\ \Lambda_B &= \mathrm{S}^\top_{x_t,m^+}\Lambda_t\mathrm{S}_{x_t,m^+} - \mathrm{S}^\top_{x_t,m^+}\Lambda_t\mathrm{S}_{m^0}\left(\mathrm{S}^\top_{m^0}\Lambda_t\mathrm{S}_{m^0}\right)^{-1}\mathrm{S}^\top_{m^0}\Lambda_t\mathrm{S}_{x_t,m^+}\end{aligned}$$

$$\begin{aligned}\boldsymbol{\eta}_C &= \mathrm{S}^\top_{m^+}\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) - \mathrm{S}^\top_{m^+}\Lambda_t\mathrm{S}_{x_t,m^0}\left(\mathrm{S}^\top_{x_t,m^0}\Lambda_t\mathrm{S}_{x_t,m^0}\right)^{-1}\mathrm{S}^\top_{x_t,m^0}\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) \\ \Lambda_C &= \mathrm{S}^\top_{m^+}\Lambda_t\mathrm{S}_{m^+} - \mathrm{S}^\top_{m^+}\Lambda_t\mathrm{S}_{x_t,m^0}\left(\mathrm{S}^\top_{x_t,m^0}\Lambda_t\mathrm{S}_{x_t,m^0}\right)^{-1}\mathrm{S}^\top_{x_t,m^0}\Lambda_t\mathrm{S}_{m^+}\end{aligned}$$

$$\begin{aligned}\boldsymbol{\eta}_D &= \mathrm{S}^\top_{m^0,m^+,m^-}\boldsymbol{\eta}_t - \mathrm{S}^\top_{m^0,m^+,m^-}\Lambda_t\mathrm{S}_{x_t}\left(\mathrm{S}^\top_{x_t}\Lambda_t\mathrm{S}_{x_t}\right)^{-1}\mathrm{S}^\top_{x_t}\boldsymbol{\eta}_t \\ \Lambda_D &= \mathrm{S}^\top_{m^0,m^+,m^-}\Lambda_t\mathrm{S}_{m^0,m^+,m^-} - \mathrm{S}^\top_{m^0,m^+,m^-}\Lambda_t\mathrm{S}_{x_t}\left(\mathrm{S}^\top_{x_t}\Lambda_t\mathrm{S}_{x_t}\right)^{-1}\mathrm{S}^\top_{x_t}\Lambda_t\mathrm{S}_{m^0,m^+,m^-}\end{aligned}$$

with $\boldsymbol{\eta}_\alpha = \Sigma_t\mathrm{S}_{m^-}\boldsymbol{\alpha}$.

3.3.2 Modified Sparsification Rule

In the previous section we showed that the SEIF derivation introduced a conditioning on a specific realization of the passive features (i.e., $\mathbf{m}^- = \boldsymbol{\alpha}$). This conditioning *influences* the outcome of the sparsification approximation and in particular can modify the resulting mean estimate as evident by the functional dependence of (3.4) on $\boldsymbol{\alpha}$. In the following we show that we can easily modify the original SEIF approximation to derive a more correct version of the sparsification rule by explicitly using the available conditional independence relationship between the robot and passive features to drop $\mathbf{m}^-$ from the posterior. This modified version of the SEIF sparsification rule preserves the state mean and, as demonstrated in §3.4, provides a high fidelity approximation that yields results comparable to the full-covariance EKF.

We begin by factorizing the posterior $p\big(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big)$ using Bayes rule just like in equation (3.2a) of the SEIF derivation, but this time we explicitly employ the available conditional independence between the robot and passive features given the active features to drop $\mathbf{m}^-$ from the posterior over $\mathbf{x}_t$ as

$$\begin{aligned}p\big(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big) &= p\big(\mathbf{x}_t \big| \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big)p\big(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big) && \text{(3.6a)} \\ &\overset{\text{C.I.}}{=} p\big(\mathbf{x}_t \big| \mathbf{m}^0, \mathbf{m}^+\big)p\big(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big) && \text{(3.6b)} \\ &= \frac{p\big(\mathbf{x}_t, \mathbf{m}^0 \big| \mathbf{m}^+\big)}{p\big(\mathbf{m}^0 \big| \mathbf{m}^+\big)}p\big(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big). && \text{(3.6c)}\end{aligned}$$

The above posterior factorization is exact, and for convenience equation (3.6c) merely re-expresses the conditional of (3.6b) as a ratio. To obtain the sparsified posterior approximation, we now impose conditional independence between $\mathbf{x}_t$ and $\mathbf{m}^0$ as

$$\begin{aligned}
\breve{p}_{\text{ModRule}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= \frac{p(\mathbf{x}_t|\mathbf{m}^+)p(\mathbf{m}^0|\mathbf{m}^+)}{p(\mathbf{m}^0|\mathbf{m}^+)} p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) && (3.7\text{a}) \\
&= p(\mathbf{x}_t|\mathbf{m}^+)p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) && (3.7\text{b}) \\
&= \frac{p_U(\mathbf{x}_t, \mathbf{m}^+)}{p_V(\mathbf{m}^+)} p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-). && (3.7\text{c})
\end{aligned}$$

For convenience, (3.7c) simplifies the sparsified posterior to a ratio of marginals where the subscripts p_U, p_V, p_D are used to notationally reference the different pdfs involved. As (3.7a)–(3.7b) show, sparsification is equivalent to imposing conditional independence, which in turn is equivalent to dropping dependence on the set of features we wish to deactivate (i.e., $\mathbf{m}^0$). The result is a modified sparsification rule summarized by equations (3.8) and (3.9) expressed in both covariance and information form respectively — see Appendix §B.3 for a full derivation.

Covariance Form

$$\begin{aligned}
\breve{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_t \\
\breve{\Sigma}_t &= \left(\mathrm{S}_{x_t,m^+}\Sigma_U^{-1}\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Sigma_V^{-1}\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^\top\right)^{-1}
\end{aligned} \qquad (3.8)$$

where

$$\begin{aligned}
\boldsymbol{\mu}_U &= \mathrm{S}_{x_t,m^+}^\top \boldsymbol{\mu}_t & \qquad \boldsymbol{\mu}_V &= \mathrm{S}_{m^+}^\top \boldsymbol{\mu}_t \\
\Sigma_U &= \mathrm{S}_{x_t,m^+}^\top \Sigma_t \mathrm{S}_{x_t,m^+} & \qquad \Sigma_V &= \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^+}
\end{aligned}$$

and $\boldsymbol{\mu}_D$, Σ_D are the same as in (3.4).

Information Form

$$\begin{aligned}
\breve{\boldsymbol{\eta}}_t &= \mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_U - \mathrm{S}_{m^+}\boldsymbol{\eta}_V + \mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D \\
\breve{\Lambda}_t &= \mathrm{S}_{x_t,m^+}\Lambda_U\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Lambda_V\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}_{m^0,m^+,m^-}^\top
\end{aligned} \qquad (3.9)$$

where

$$\begin{aligned}
\boldsymbol{\eta}_U &= \mathrm{S}_{x_t,m^+}^\top\boldsymbol{\eta}_t - \mathrm{S}_{x_t,m^+}^\top\Lambda_t\mathrm{S}_{m^0,m^-}\left(\mathrm{S}_{m^0,m^-}^\top\Lambda_t\mathrm{S}_{m^0,m^-}\right)^{-1}\mathrm{S}_{m^0,m^-}^\top\boldsymbol{\eta}_t \\
\Lambda_U &= \mathrm{S}_{x_t,m^+}^\top\Lambda_t\mathrm{S}_{x_t,m^+} - \mathrm{S}_{x_t,m^+}^\top\Lambda_t\mathrm{S}_{m^0,m^-}\left(\mathrm{S}_{m^0,m^-}^\top\Lambda_t\mathrm{S}_{m^0,m^-}\right)^{-1}\mathrm{S}_{m^0,m^-}^\top\Lambda_t\mathrm{S}_{x_t,m^+}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\eta}_V &= \mathrm{S}_{m^+}^\top\boldsymbol{\eta}_t - \mathrm{S}_{m^+}^\top\Lambda_t\mathrm{S}_{x_t,m^0,m^-}\left(\mathrm{S}_{x_t,m^0,m^-}^\top\Lambda_t\mathrm{S}_{x_t,m^0,m^-}\right)^{-1}\mathrm{S}_{x_t,m^0,m^-}^\top\boldsymbol{\eta}_t \\
\Lambda_V &= \mathrm{S}_{m^+}^\top\Lambda_t\mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top\Lambda_t\mathrm{S}_{x_t,m^0,m^-}\left(\mathrm{S}_{x_t,m^0,m^-}^\top\Lambda_t\mathrm{S}_{x_t,m^0,m^-}\right)^{-1}\mathrm{S}_{x_t,m^0,m^-}^\top\Lambda_t\mathrm{S}_{m^+}
\end{aligned}$$

and $\boldsymbol{\eta}_D$, Λ_D are the same as in (3.5).

In particular, note that equation (3.8) shows that the modified-rule clearly maintains the mean estimate. Furthermore, careful inspection of the projection matrices involved in (3.9) shows that it simultaneously deactivates the map features $\mathbf{m}^0$ (i.e., S_{x_t,m^+} only populates the robot/active feature sub-block of the resulting information matrix $\breve{\Lambda}_t$). However, a significant drawback (despite its correctness) is that sparsification is no longer a constant-time operation as evident by the expressions for Λ_U and Λ_V which require large matrix inversions over the passive features $\mathbf{m}^-$.

3.3.3 What happens if we just leave $\mathbf{m}^-$ alone?

In §3.3.1 we showed that conditioning the SEIF posterior (3.3) on $\mathbf{m}^- = \boldsymbol{\alpha}$ influences the approximation's outcome which is an undesirable attribute. However, its advantage is the fact that computing $\tilde{\boldsymbol{\eta}}$, $\tilde{\Lambda}$ is a constant-time operation since only a matrix of the size of the deactivated features $\mathbf{m}^0$ needs to be inverted. Section 3.3.2 then derived a modified sparsification rule that exploits the conditional independence between $\mathbf{x}_t$ and $\mathbf{m}^-$ to yield a sparsification rule that preserves the distribution's mean while simultaneously deactivating $\mathbf{m}^0$. Unfortunately, this rule is no longer constant-time (in fact cubic) because it requires inverting a matrix of the size of the passive features $\mathbf{m}^-$ (this constitutes a majority of the map). Therefore, a natural question to ask is what happens if we leave $\mathbf{m}^-$ alone in the conditioning? In other words, suppose that the sparsification step merely drops the dependence of $\mathbf{x}_t$ on $\mathbf{m}^0$ while leaving $\mathbf{m}^-$ in the conditioning — what will the result be?

We begin by factorizing the original distribution using Bayes rule as

$$\begin{aligned}
p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= p(\mathbf{x}_t | \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) && (3.10\text{a}) \\
&= \frac{p(\mathbf{x}_t, \mathbf{m}^0 | \mathbf{m}^+, \mathbf{m}^-)}{p(\mathbf{m}^0 | \mathbf{m}^+, \mathbf{m}^-)} p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-). && (3.10\text{b})
\end{aligned}$$

Next we drop $\mathbf{x}_t$'s dependence on $\mathbf{m}^0$ by forcing $\mathbf{x}_t$ and $\mathbf{m}^0$ to be conditionally independent given $\mathbf{m}^+$ and $\mathbf{m}^-$ as

$$\begin{aligned}
\hat{p}\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) &= \frac{p(\mathbf{x}_t | \mathbf{m}^+, \mathbf{m}^-) p(\mathbf{m}^0 | \mathbf{m}^+, \mathbf{m}^-)}{p(\mathbf{m}^0 | \mathbf{m}^+, \mathbf{m}^-)} p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) && (3.11\text{a}) \\
&= p(\mathbf{x}_t | \mathbf{m}^+, \mathbf{m}^-) p(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-). && (3.11\text{b})
\end{aligned}$$

After working through the math the resulting distribution in canonical form is given by

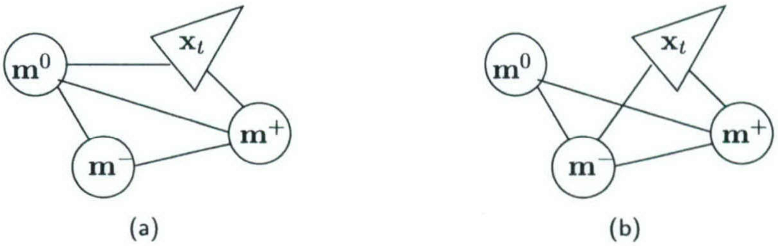
$$\begin{aligned}
\hat{\boldsymbol{\eta}}_t &= \mathrm{S}_{x_t,m^+,m^-} \boldsymbol{\eta}_P - \mathrm{S}_{m^+,m^-} \boldsymbol{\eta}_Q + \mathrm{S}_{m^0,m^+,m^-} \boldsymbol{\eta}_D \\
\hat{\Lambda}_t &= \mathrm{S}_{x_t,m^+,m^-} \Lambda_P \mathrm{S}_{x_t,m^+,m^-}^\top - \mathrm{S}_{m^+,m^-} \Lambda_Q \mathrm{S}_{m^+,m^-}^\top + \mathrm{S}_{m^0,m^+,m^-} \Lambda_D \mathrm{S}_{m^0,m^+,m^-}^\top
\end{aligned} \qquad (3.12)$$

where

$$\begin{aligned}
\boldsymbol{\eta}_P &= \mathrm{S}_{x_t,m^+,m^-}^\top \boldsymbol{\eta}_t - \mathrm{S}_{x_t,m^+,m^-}^\top \Lambda_t \mathrm{S}_{m^0} \left(\mathrm{S}_{m^0}^\top \Lambda_t \mathrm{S}_{m^0}\right)^{-1} \mathrm{S}_{m^0}^\top \boldsymbol{\eta}_t \\
\Lambda_P &= \mathrm{S}_{x_t,m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^+,m^-} - \mathrm{S}_{x_t,m^+,m^-}^\top \Lambda_t \mathrm{S}_{m^0} \left(\mathrm{S}_{m^0}^\top \Lambda_t \mathrm{S}_{m^0}\right)^{-1} \mathrm{S}_{m^0}^\top \Lambda_t \mathrm{S}_{x_t,m^+,m^-}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\eta}_Q &= \mathrm{S}_{m^+,m^-}^\top \boldsymbol{\eta}_t - \mathrm{S}_{m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^0} \left(\mathrm{S}_{x_t,m^0}^\top \Lambda_t \mathrm{S}_{x_t,m^0}\right)^{-1} \mathrm{S}_{x_t,m^0}^\top \boldsymbol{\eta}_t \\
\Lambda_Q &= \mathrm{S}_{m^+,m^-}^\top \Lambda_t \mathrm{S}_{m^+,m^-} - \mathrm{S}_{m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^0} \left(\mathrm{S}_{x_t,m^0}^\top \Lambda_t \mathrm{S}_{x_t,m^0}\right)^{-1} \mathrm{S}_{x_t,m^0}^\top \Lambda_t \mathrm{S}_{m^+,m^-}
\end{aligned}$$

Figure 3-4 Leaving $\mathbf{m}^-$ alone in the conditioning reactivates passive features. (a) The Markov network associated with the SEIF posterior. (b) The Markov network after sparsification by (3.10). The approximation forces $\mathbf{x}_t$ and $\mathbf{m}^0$ to be conditionally independent given $\mathbf{m}^- \cup \mathbf{m}^+$, but doesn't constrain $\mathbf{x}_t$ and $\mathbf{m}^-$ to maintain a sparse relationship.



and $\boldsymbol{\eta}_D$, Λ_D are the same as in (3.5).

Close inspection of the sparsified information matrix, $\hat{\Lambda}_t$, shows that we have successfully deactivated $\mathbf{m}^0$ by setting the shared information between $\mathbf{x}_t$ and $\mathbf{m}^0$ to zero (i.e., none of the projection matrices in (3.12) extract them together). However, further inspection of (3.12) shows that, in practice, this approximation is not very useful because the first term in the expression (i.e., $\mathrm{S}_{x_t,m^+,m^-}\Lambda_P\mathrm{S}^\top_{x_t,m^+,m^-}$) introduces *new* links into the information matrix. These links occur between the robot, $\mathbf{x}_t$, the active features, $\mathbf{m}^+$, and the passive feature, $\mathbf{m}^-$, and will destroy our sparsity at the next time-projection step. Intuitively, the sparsified posterior (3.11a) forces $\mathbf{x}_t$ and $\mathbf{m}^0$ to be conditionally independent given $\mathbf{m}^+ \cup \mathbf{m}^-$, but doesn't enforce any type of sparsity constraint between $\mathbf{x}_t$ and $\mathbf{m}^-$. Therefore, the Markov network is allowed to reorganize itself into the sparsified graph depicted in Fig. 3-4. So while we managed to remove a small number of links from the robot, $\mathbf{x}_t$, to the deactivated features, $\mathbf{m}^0$, we have reactivated many more links between it and the passive features, $\mathbf{m}^-$. As a result, these reactivated links will now cause the information matrix to densify during the next robot motion time-prediction (§3.2.3).

3.4 Results

This section investigates the implications of the two different sparsification rules by comparing the results of the sparsified information filters to that of the standard Kalman formulation.[4] In the first scenario, §3.4.1, we consider a linear Gaussian (LG) SLAM simulation where the Kalman filter (KF) *is* the optimal Bayes estimator and provides a benchmark measure against which to compare the different sparsification routines. Subsequently, in the second scenario, §3.4.2, we test the algorithms on a nonlinear real-world dataset to understand their performance in practice.

[4] These results were produced jointly with Matthew Walter [39].

3.4.1 Simulation: LG SLAM

Experimental Setup

In an effort to compare the effects of the two sparsification strategies in a controlled manner, we applied the three estimators (i.e., SEIF, modified-rule, and KF) to a synthetic, linear, 2D dataset. For this simulation, vehicle motion was constrained to be purely translational (fixed orientation) and was generated by a linear, constant-velocity process model corrupted by additive white Gaussian noise. As the robot moved around in the environment, it measured the relative position of local point features, again perturbed by white noise, and had an active feature bound set at 10% of the total number of landmarks in the environment. Since the simulation was restricted to LG SLAM, we can compare the effects of the two sparsification routines relative to the optimal (KF) solution.

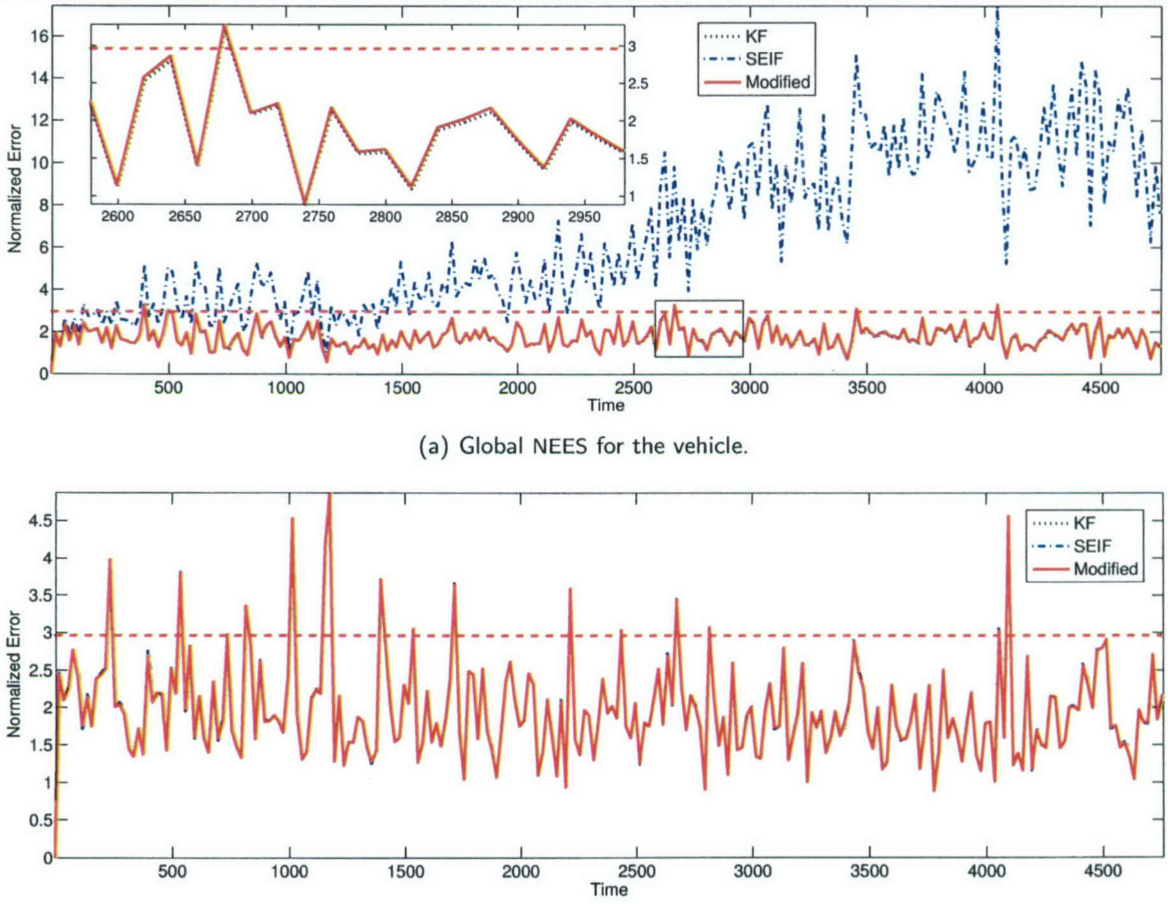
Experimental Results

To test the consistency between the different filter covariances and the true state estimation errors, we use the normalized estimation error squared (NEES) [7, §5.4.2] as computed based upon a series of Monte Carlo simulations and two different error metrics. The first metric compares the direct output of the filters to ground-truth and provides a measure of *global* error. The second metric computes the state estimate relative to the first feature that was observed, denoted $\mathbf{x}_m$, via the standard compounding operation: $\mathbf{x}_{mi} = \ominus\mathbf{x}_m \oplus \mathbf{x}_i$ [154], which provides a measure of *local/relative* error. Fig. 3-5 compares the two error metrics for the vehicle position NEES score for the KF and information filters. Similarly, Fig. 3-6 shows the normalized errors for a single map feature and is representative of the performance for other map elements. The horizontal threshold in both plots signifies the 97.5% upper bound for the chi-square test. Comparing the estimate of vehicle and map positions in the world frame, the modified-rule yields errors nearly identical to those of the KF, not only in regards to magnitude, but also in behavior over time. In comparison, the SEIF global errors are noticeably larger, though in contrast, the normalized relative errors are roughly equivalent to those of the KF and modified-rule. This apparent discrepancy indicatives that the relative map estimates for all three filters have converged while the global SEIF estimate is inconsistent.

We gain further insight into the consequences of sparsification by looking at the covariances associated with each filter. Fig. 3-7 depicts a histogram comparing the ratio of determinants for the global and relative KF feature covariances to those of the two information filters. To aid in interpreting the ratio as a metric, values of one represent ideal, while those larger than one indicate the amount of overconfidence. For both information filters, the uncertainty measures are overconfident with respect to those of the optimal KF, and in turn are inconsistent with the true estimation error. However, the difference in magnitude between the modified-rule's confidence regions to those of the standard KF are nearly negligible in both a global and local sense. In contrast the SEIF rule has absolute uncertainty that is significantly more overconfident.[5] Meanwhile, upon referencing the state estimates relative to the first observed feature, the SEIF covariance matrix now reflects nearly the

[5] The exception is with the first feature added to the map which is the source of the outliers shown in the plots in Fig. 3-7(a).

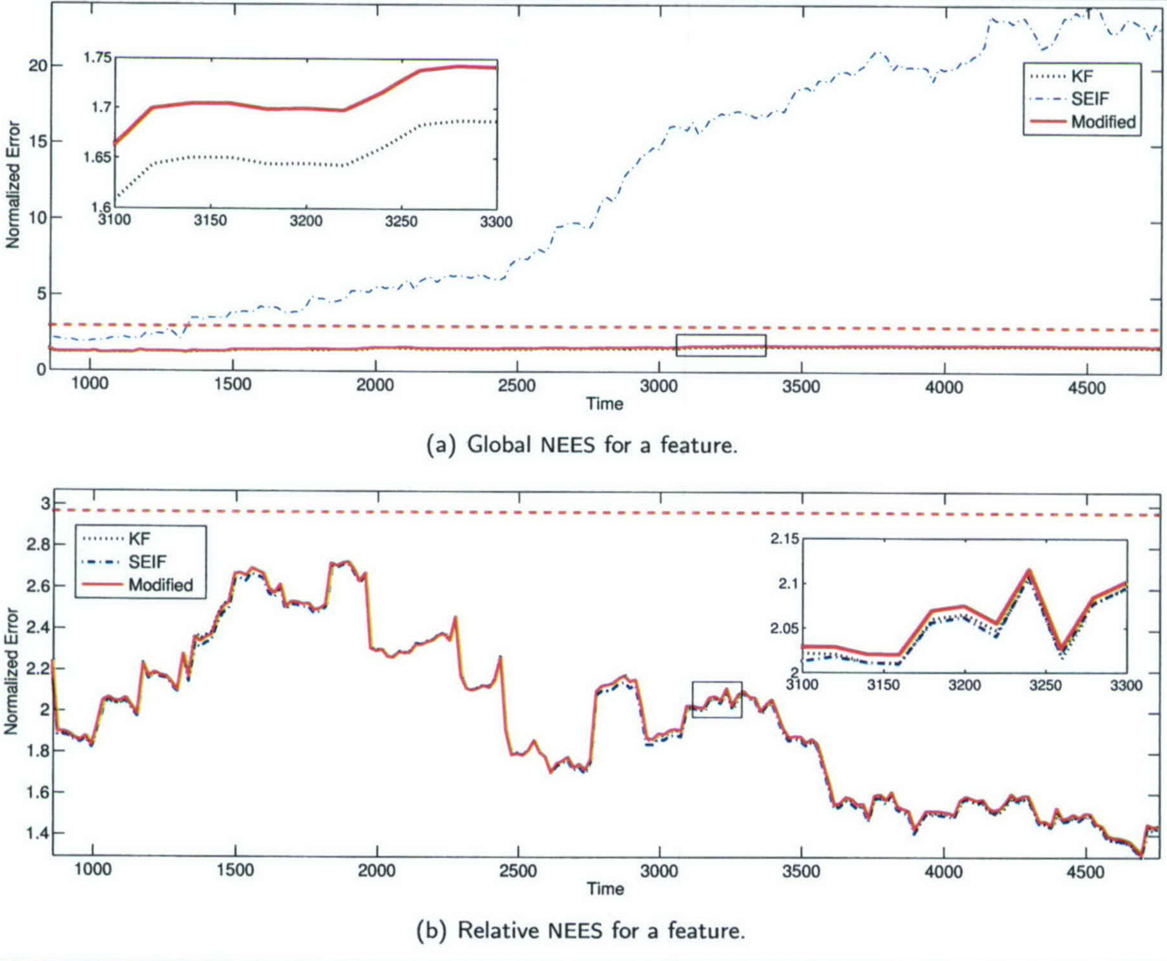
Figure 3-5 The normalized estimation error squared (NEES) for the vehicle as computed based upon 20 linear Gaussian Monte Carlo simulations [39]. The horizontal dotted-line signifies the the 97.5% chi-square upper bound in both plots. The error shown in (a) corresponds to a direct comparison of the filter estimates to the ground-truth, and represents a measure of *global* consistency. In (b) we plot the *local* normalized error computed relative to the first feature instantiated in the map (i.e., $\mathbf{x}_{mi} = \ominus\mathbf{x}_m \oplus \mathbf{x}_i$). SEIF's global normalized error is larger as a result of an absolute state that is significantly overconfident. On the other hand, the relative map error is nearly identical to that of the modified-rule and KF, which empirically indicates that the SEIF yields locally consistent estimates.



(a) Global NEES for the vehicle.

(b) Relative NEES for the vehicle.

Figure 3-6 The NEES for a representative feature. See the caption of Fig. 3-5 for a description.



(a) Global NEES for a feature.

(b) Relative NEES for a feature.

same level of uncertainty as the KF and modified-rule. In the process of root-shifting the map to the first feature, the original world origin now becomes included as a state element. Note that while the world origin uncertainty estimate for the modified-rule agrees with those of the rest of the relative map estimates, the same is not true for the SEIF's uncertainty measure of the world-origin as indicated by the outlier in Fig. 3-7(b). This suggests that while the relative SEIF map estimate has converged, its estimate of the global world origin remains inconsistent.

The effect of sparsification on the covariance estimates is consistent with what is observed with the normalized errors. In other words, though there is little difference between the three sets of feature position estimates, the errors for the global SEIF map are much larger due to overconfidence in its state estimates. However upon root-shifting, the difference in error between the different relative maps becomes negligible.

3.4.2 Experimental Validation: MIT Tennis Courts

While linear simulations are helpful for investigating our findings, more often than not, real-world SLAM problems involve nonlinear vehicle motion and perception models. Furthermore, real data often includes noise that is not truly Gaussian. For these reasons, we tested the estimation algorithms on a typical, nonlinear dataset.

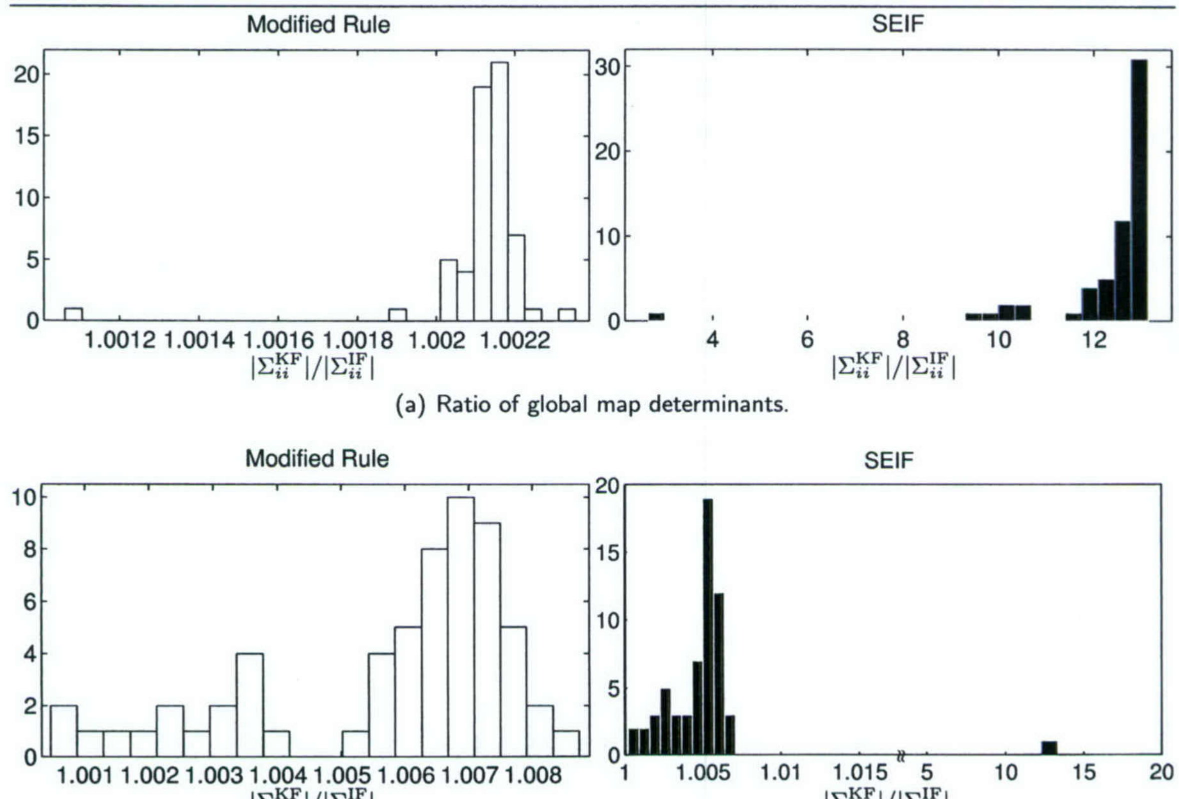
Experimental Setup

As depicted in Fig. 3-8, a wheeled robot was manually driven around an environment consisting of 64 track hurdles positioned on four adjacent tennis courts, which provide ground-truth. The vehicle observed the environment using a SICK laser range finder and was equipped with wheel encoders for determining the motion control inputs. Data association (the problem of correctly pairing measurement data with the corresponding hurdle) was addressed offline [167] and, thus, is identical for each SLAM filter. In our feature-based representation, each hurdle serves as an individual coordinate frame parameterized by a base leg position and its orientation.

Experimental Results

An EKF was applied alongside the two different information filters who each had an active feature bound of 10 landmarks. The resulting SLAM maps are shown in Fig. 3-9 and are seen to exhibit very similar behavior to the LG SLAM results (i.e., that SEIF has a contrasting global/relative performance). Fig. 3-9(a) plots the SLAM maps in terms of the global state representation where we see that enforcing sparsity with the modified-rule leads to a negligible difference from the EKF. However, like in the LG simulation, the SEIF yields global map estimates that are inconsistent, since a majority of the true hurdle positions fall well outside the 3-sigma uncertainty regions. Meanwhile, as depicted in Fig. 3-9(b), root-shifting the map relative to the first instantiated feature yields a relative posterior very similar to the EKF.

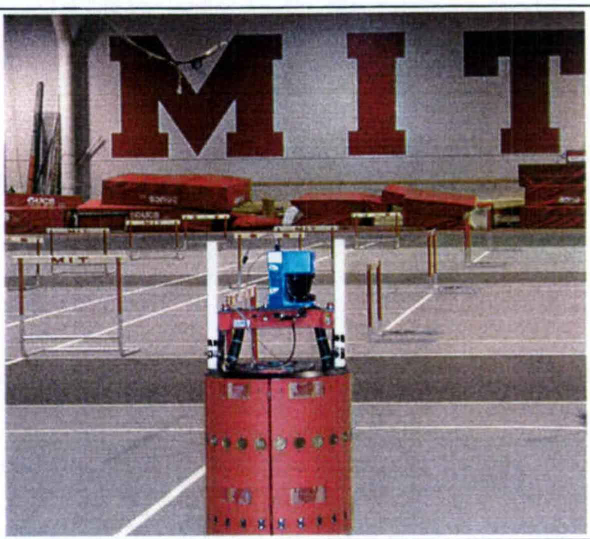
Figure 3-7 LG SLAM simulation: comparison of the SEIF and modified-rule filter covariances to the optimal KF. For each map element we compute the ratio of the KF feature determinant to the information filter (IF) feature determinant — hence, ratios greater than 1 indicate by how much each IF is overconfident. To facilitate comparison of all map elements, these ratios are displayed as histograms. (a) The uncertainties are computed directly from the covariances maintained by the filters. Note that the outlier in both histograms (entry nearest to one) corresponds to the first mapped feature. (b) The uncertainties are computed for a *relative* map w.r.t. to the first observed feature. Though both sparsification rules are overconfident, note that the modified-rule is nearly equivalent to the KF (ratios ≈ 1) for both the global and relative maps. Meanwhile, the global SEIF estimates are significantly overconfident while the relative estimates are approximately equivalent to the KF. In addition, the outlier in the SEIF histogram corresponds to the original world origin as represented in the root-shifted reference frame — its significant overconfidence is a direct consequence of the overconfident absolute map.



(a) Ratio of global map determinants.

(b) Ratio of relative map determinants.

Figure 3-8 The experimental setup used to collect the hurdles dataset on the MIT tennis court. Ground-truth is determined from the court baselines.



3.5 Discussion

While the modified-rule was shown to outperform the SEIF by producing error estimates nearly identical in both a global and local sense to those of the standard Kalman formulation, close inspection reveals that its estimates are still slightly overconfident with respect to the KF. This section seeks to explain the cause of this inconsistency.

3.5.1 Imposing Conditional Independence Leads to Inconsistency

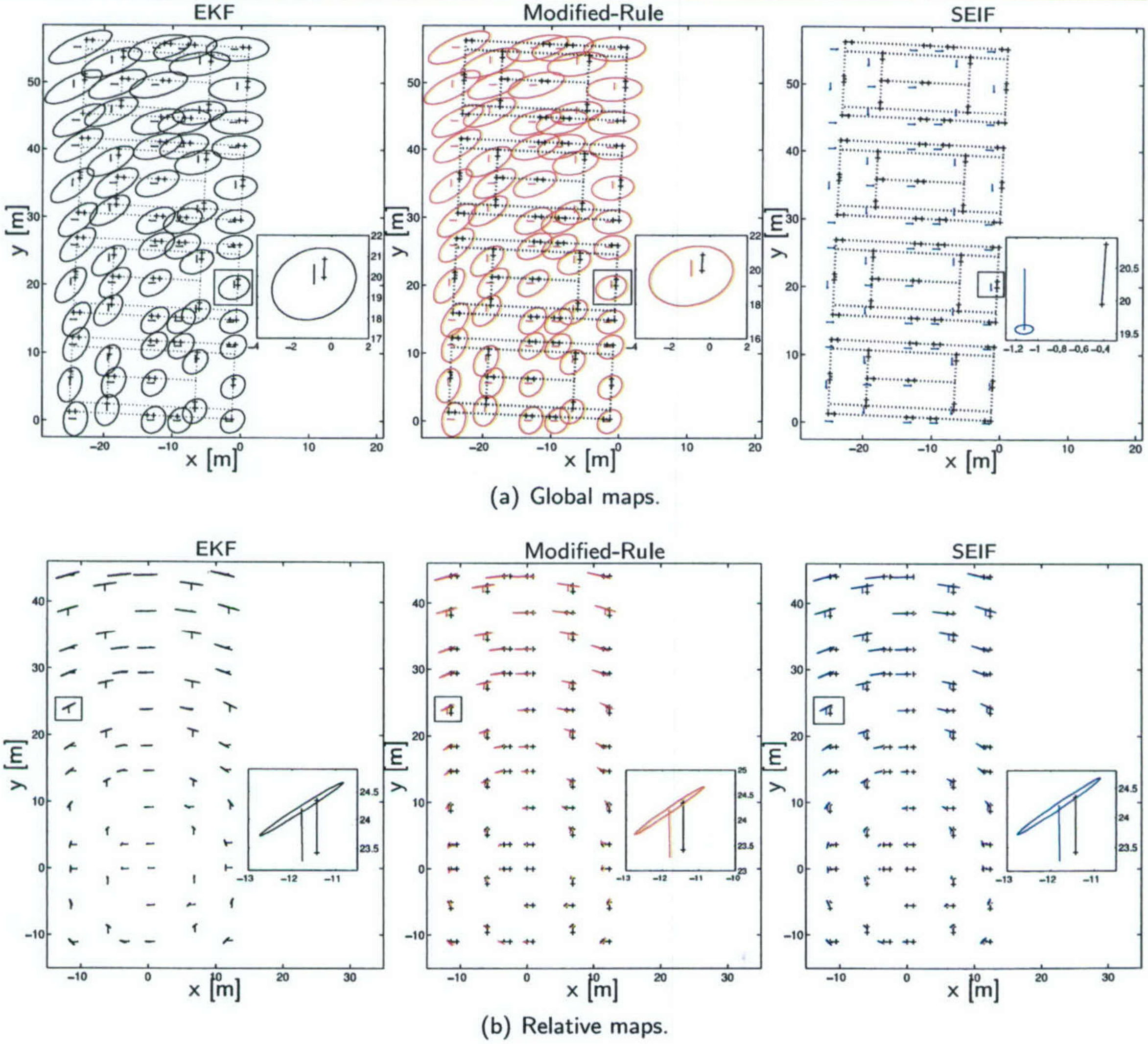
Instructively, it can be shown that the overconfidence is a direct result of *imposing* conditional independence between the robot and the deactivated features, $\mathbf{m}^0$. To illustrate this, consider the general three state distribution $p(a, b, c) = p(a|b, c)p(b, c)$, and its sparsified approximation where any possible dependence between a and b is ignored:

$$\tilde{p}(a, b, c) = p(a|c)p(b, c). \tag{3.13}$$

To understand the effect of this approximation on LG SLAM, suppose that the true distribution in covariance form is given by

$$p(a, b, c) = \mathcal{N}\left(\begin{bmatrix} \mu_a \\ \mu_b \\ \mu_c \end{bmatrix}, \begin{bmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b & \rho_{ac}\sigma_a\sigma_c \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 & \rho_{bc}\sigma_b\sigma_c \\ \rho_{ac}\sigma_a\sigma_c & \rho_{bc}\sigma_b\sigma_c & \sigma_c^2 \end{bmatrix} \right) \tag{3.14}$$

Figure 3-9 Comparison of the EKF and information filter SLAM maps with ground-truth (cross-hairs) for the hurdles experimental dataset (from Eustice, Walter, and Leonard [39]). The plots in (a) correspond to the global feature poses as directly estimated by the three SLAM algorithms together with the 3-sigma confidence bounds. Shown in (b) are the relative maps and corresponding 3-sigma uncertainty ellipses transformed relative to the first hurdle added to the map. As indicated by the right-most plot of (a), the SEIF maintains global feature estimates that are significantly overconfident. Meanwhile, the modified-rule yields estimates for global feature pose and uncertainty that are nearly identical to those of the EKF. Considering the relative maps (b), the two sparsified filters perform similarly to the EKF.



(a) Global maps.

(b) Relative maps.

where ρ_{ab}, ρ_{ac}, and ρ_{bc} are the normalized correlation coefficients. Applying the sparsification approximation of (3.13) yields

$$\tilde{p}(a,b,c) = \mathcal{N}\left(\begin{bmatrix}\mu_a\\ \mu_b\\ \mu_c\end{bmatrix}, \begin{bmatrix}\sigma_a^2 & \rho_{ac}\rho_{bc}\sigma_a\sigma_b & \rho_{ac}\sigma_a\sigma_c\\ \rho_{ac}\rho_{bc}\sigma_a\sigma_b & \sigma_b^2 & \rho_{bc}\sigma_b\sigma_c\\ \rho_{ac}\sigma_a\sigma_c & \rho_{bc}\sigma_b\sigma_c & \sigma_c^2\end{bmatrix}\right). \tag{3.15}$$

A necessary and sufficient condition for consistency is that the covariance matrices must obey the inequality, $\tilde{\Sigma} - \Sigma \geq 0$ [23]. To test whether or not this condition is true for (3.15), we note that a sufficient condition test for positive semi-definiteness is that the determinant of all upper left sub-matrices must be positive [156]. Applying this test to (3.15) we get

$$\tilde{\Sigma} - \Sigma = \begin{bmatrix}0 & (\rho_{ac}\rho_{bc} - \rho_{ab})\sigma_a\sigma_b & 0\\ (\rho_{ac}\rho_{bc} - \rho_{ab})\sigma_a\sigma_b & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \ngeq 0 \tag{3.16}$$

where in general the determinant of the upper left 2×2 of (3.16) is less than zero for $\rho_{ac}\rho_{bc} \neq \rho_{ab}$. Hence, extending this insight to both the modified and original SEIFs rule, we see that imposing conditional independence between the robot and the deactivated features results in an approximation that is inconsistent. Therefore, while the modified-rule estimates are comparable to the KF, this explains the cause of their slight overconfidence.

3.6 Chapter Summary

In conclusion, recent novel insights into the canonical formulation of SLAM have revealed that sparseness is a "natural" characteristic of the information parameterization. This has lead to promising new research into scalable algorithms based upon pruning relatively weak constraints in the information form to achieve sparsity. The delicate issue, however, is "how to approximate the posterior with an exactly sparse representation in a consistent manner?"

In this chapter we demonstrated that the constant-time SEIF method of enforcing sparsity leads to an inconsistent *global* map, however, empirical testing indicates that the *relative* map relationships are preserved. We also showed that by exploiting the conditional independence between the robot and the passive features given the active map, that a new alternative version of the SEIF sparsification rule can be derived. This modified-rule yields a sparsified posterior comparable to that of the EKF for both global and relative maps, but unfortunately this accuracy comes at an impractical cost since it requires matrix inversion over the passive features, which implies cubic-time.

In addtion, we also noted that LG SLAM simulation results indicated that sparsification leads to an overconfident state estimate. We investigated the cause of this inconsistency in §3.5 and concluded that this overconfidence is a direct result of imposing conditional independence between state variables. Hence, this suggests that both a consistent and computationally efficient approximation for imposing sparsity in feature-based SLAM remains an open research task. Insightfully, Chapter 4 will show that there exists an alternative case where exact sparsity can be achieved without approximation, and that this case is view-based SLAM.

CHAPTER 4

Exactly Sparse Delayed-State Filters

This chapter presents the novel insight that the SLAM information matrix is exactly sparse for a *view-based* framework. A view-based representation relies upon scan-matching raw sensor data and maintaining an associated collection of delayed robot states such that registration results in virtual observations of robot motion with respect to a place it has previously been (Chapter 2). Within this framework, the exact sparsity of the delayed-state information matrix is in contrast to the approximate sparsity of other recent feature-based SLAM information algorithms (Chapter 3). Furthermore, this exact sparsity is a direct result of the Markov property of the process model coupled with the fact that the Gaussian canonical form encodes *constraints* between random variables — we call this result "exactly sparse delayed-state filters (ESDFs)". The benefit is that a delayed-state framework can take advantage of the sparse information matrix parameterization without having to incur any approximation error. Therefore, we can use ESDFs for our VAN methodology to produce equivalent results to the "full-covariance" EKF solution, however, at only $\mathcal{O}(n)$ cost.

Another novel contribution that we present is a clever strategy for efficiently accessing and maintaining consistent marginal covariances within a SLAM information filter, thereby greatly increasing the reliability of data association. Since naïve access to the covariance matrix requires matrix inversion (a cubic operation), we have developed a novel technique based upon solving a sparse system of linear equations coupled with the application of constant-time Kalman updates. Essentially, our strategy maintains estimates of the covariance block-diagonal and, in addition, the robot's cross-correlation to these elements. As we show, this technique produces consistent covariance estimates suitable for robot planning and data association — something that has been an open research issue for all SLAM information filters.

4.1 Background

To our knowledge, the earliest related work that exploited the efficiency of the measurement update in the inverse covariance form was published by McLauchlan and Murray [103], in the context of recursive structure-from-motion (SFM). This work was subsequently extended to realize a hybrid batch/recursive visual SLAM implementation that unified recursive SLAM and bundle adjustment [105]. McLauchlan recognized the potential increase in efficiency that can be gained via approximations to maintain sparsity of the information matrix:

> It has long been known in the photogrammetry community, in the form of the equivalent normal formulation, that the [information] matrix ... takes a special sparse form in the context of reconstruction .... [However, in a recursive formulation] ... eliminating motion fills in the structure blocks. This has to be avoided to maintain update times proportional to n. So our *partial elimination adjustment* method is to ignore corrections that fill-in zero blocks, while applying the correction to the blocks which are already non-zero.

While the consistency implications of this approximation are unknown, in practice the method achieved results approaching those of a full batch solution for moderate duration image sequences.

Recently, the SLAM community has also turned its attention to exploring the information parameterization for increased efficiency. In particular, Chapter 3 discussed a new class of scalable feature-based SLAM algorithms founded upon representing the posterior in the canonical form. Since the information form naturally yields an "almost sparse" representation (whereby the robot-landmark information matrix is dominated by a relatively few number of entries), these new algorithms achieve scalability by eliminating the weak entries and exploiting the remaining sparse representation (e.g., sparse extended information filters (SEIFs) [160], Thin Junction-Tree Filters [121], and Treemap Filters [49]). However, *enforcing* sparsity is necessarily an *approximation*, which (as we saw in the case of SEIFs) can lead to map estimate inconsistency.

Interestingly, it is the same phenomenon that plagues both the information formulations of McLauchlan and Murray [103, 105], and the feature-based SLAM algorithms of Thrun et al. [160], Paskin [121], and Frese [49] — and that is that "eliminating motion fills in the structure blocks." In §3.2.3 we discussed this concept in depth and illustrated how eliminating the robot's trajectory causes the SLAM landmark posterior to densify. This fill-in destroys sparsity and, hence, any resulting efficiency associated with a sparse representation. This is the reason why *all* feature-based SLAM information algorithms are founded upon some type of pruning strategy that removes weak constraints. Fortunately, because we do maintain samples from the robot's trajectory in our VAN framework, and because the information matrix represents constraints among random variables, our view-based representation is intrinsically sparse. Hence, VAN can exploit the efficiency of SLAM information filters in a consistent manner.

In the following, we present both our view-based SLAM information framework (§4.2) and our conservative data association technique (§4.3). Benchmark results (§4.5) quantifying the view-based information filter efficiency with respect to the standard EKF VAN formulation of Chapter 2 are shown for the JHU dataset. In addition, real-world results are presented for the largest visually-navigated underwater dataset to date using data from a recent ROV survey of the wreck of the RMS Titanic.

4.2 Filter Mechanics

4.2.1 State Augmentation

We begin by describing the method of state augmentation, which is how we "grow" the state vector to contain a new delayed-state. This operation occurs whenever we have a new "view" that we wish to store. For example, in our VAN framework we add a delayed-state for each acquired image of the environment that we wish to be able to revisit at a later time.

Adding a Delayed-State

Assume for the moment that our estimate at time t is described by the distribution given below expressed in both covariance and information form.

$$p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^t) = \mathcal{N}\Big(\begin{bmatrix}\boldsymbol{\mu}_{x_t}\\ \boldsymbol{\mu}_M\end{bmatrix}, \begin{bmatrix}\Sigma_{x_t x_t} & \Sigma_{x_t M}\\ \Sigma_{M x_t} & \Sigma_{MM}\end{bmatrix}\Big) = \mathcal{N}^{-1}\Big(\begin{bmatrix}\boldsymbol{\eta}_{x_t}\\ \boldsymbol{\eta}_M\end{bmatrix}, \begin{bmatrix}\Lambda_{x_t x_t} & \Lambda_{x_t M}\\ \Lambda_{M x_t} & \Lambda_{MM}\end{bmatrix}\Big)$$

This distribution represents a map and current robot state, $\mathbf{M}$ and $\mathbf{x}_t$ respectively, given all measurements, $\mathbf{z}^t$, and control inputs, $\mathbf{u}^t$. Here the map variable $\mathbf{M}$ is used in a general sense, for example it could represent a collection of delayed-states or a set of landmark features in the environment. For now we don't care, because we want to show what happens when we augment our representation to include the time-propagated robot state, $\mathbf{x}_{t+1}$, obtaining the distribution $p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1})$, which can be factored as

$$\begin{aligned} p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1}) &= p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{M}, \mathbf{z}^t, \mathbf{u}^{t+1}) p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1}) \\ &\overset{\text{Markov}}{=} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_{t+1}) p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^t). \end{aligned} \tag{4.1}$$

In (4.1) we factored the posterior into the product of a probabilistic state-transition multiplied by our prior using the common assumption that the robot state evolves according to a first-order Markov process. Equation (4.2) describes the general nonlinear discrete-time Markov robot motion model we assume and (4.3) its first-order linearized form where F is the Jacobian evaluated at $\boldsymbol{\mu}_{x_t}$ and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{Q})$ is the white process noise.[1]

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_{t+1}) + \mathbf{w}_t \tag{4.2}$$

$$\approx \mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) + \mathrm{F}(\mathbf{x}_t - \boldsymbol{\mu}_{x_t}) + \mathbf{w}_t \tag{4.3}$$

Note that in general our robot state description, $\mathbf{x}_t$, consists of both pose (i.e., position and orientation) *and* kinematic components (e.g., body-frame velocities, angular rates).

Augmentation in the Covariance Form

Under the linearized approximation of (4.3), the augmented state distribution of (4.1) is also Gaussian, and in covariance form its result is given by [155]:

$$p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1}) = \mathcal{N}(\boldsymbol{\mu}'_{t+1}, \Sigma'_{t+1})$$

[1] See Appendix §A.3 for a description of our discrete-time vehicle model.

$$\boldsymbol{\mu}'_{t+1} = \begin{bmatrix} \mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) \\ \boldsymbol{\mu}_{x_t} \\ \boldsymbol{\mu}_M \end{bmatrix} \qquad \Sigma'_{t+1} = \begin{bmatrix} (\mathrm{F}\Sigma_{x_tx_t}\mathrm{F}^\top + \mathrm{Q}) & \mathrm{F}\Sigma_{x_tx_t} & \mathrm{F}\Sigma_{x_tM} \\ \Sigma_{x_tx_t}\mathrm{F}^\top & \Sigma_{x_tx_t} & \Sigma_{x_tM} \\ \Sigma_{Mx_t}\mathrm{F}^\top & \Sigma_{Mx_t} & \Sigma_{MM} \end{bmatrix}. \tag{4.4}$$

The lower-right 2×2 sub-block of Σ'_{t+1} corresponds to the covariance between the delayed robot state element, $\mathbf{x}_t$, and the map, $\mathbf{M}$, and has remained unchanged from the prior. Meanwhile, the first row and column contain the cross-covariances associated with the time propagated robot state, $\mathbf{x}_{t+1}$, which includes the effect of the process model.

Augmentation in the Information Form

Having obtained the delayed-state distribution in covariance form, we can now transform (4.4) to its information form (4.5). This requires inversion of the 3×3 block covariance matrix Σ'_{t+1} whose tedious derivation we omit here, though, note that (4.5) can be verified by the fact that $\Lambda'_{t+1}\Sigma'_{t+1} = \mathrm{I}$ and $\boldsymbol{\eta}'_{t+1} = \Lambda'_{t+1}\boldsymbol{\mu}'_{t+1}$.

$$p\big(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} \big| \mathbf{z}^t, \mathbf{u}^{t+1}\big) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}'_{t+1}, \Lambda'_{t+1}\big)$$

$$\boldsymbol{\eta}'_{t+1} = \begin{bmatrix} \mathrm{Q}^{-1}\big(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big) \\ \boldsymbol{\eta}_{x_t} - \mathrm{F}^\top\mathrm{Q}^{-1}\big(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big) \\ \boldsymbol{\eta}_M \end{bmatrix} \qquad \Lambda'_{t+1} = \begin{bmatrix} \mathrm{Q}^{-1} & -\mathrm{Q}^{-1}\mathrm{F} & \boxed{0} \\ -\mathrm{F}^\top\mathrm{Q}^{-1} & (\Lambda_{x_tx_t} + \mathrm{F}^\top\mathrm{Q}^{-1}\mathrm{F}) & \Lambda_{x_tM} \\ \underset{\text{key result}}{\boxed{0}} & \Lambda_{Mx_t} & \Lambda_{MM} \end{bmatrix} \tag{4.5}$$

Markovity Yields Exact Sparseness

Equation (4.5) yields a key insight into the structure of the information matrix regarding delayed-states. We see that augmenting our state vector to include the time-propagated robot state, $\mathbf{x}_{t+1}$, introduces shared information only between it and the previous robot state, $\mathbf{x}_t$. Moreover, the shared information between $\mathbf{x}_{t+1}$ and the map, $\mathbf{M}$, is *always* zero irrespective of what $\mathbf{M}$ abstractly represents (i.e., regardless of whether $\mathbf{M}$ represents a set of landmarks or a collection of delayed-states, the result will always be zero). This sparsity in the augmented state information matrix is a consequence of the Markov property associated with the state transition probability $p\big(\mathbf{x}_{t+1} \big| \mathbf{x}_t, \mathbf{u}_{t+1}\big)$. In terms of a Markov network [122], $\mathbf{x}_{t+1}$ is only serially connected to its parent node, $\mathbf{x}_t$, and therefore, is conditionally independent of $\mathbf{M}$.

By induction, a key property of state augmentation in the information form is that if we continue to augment our state vector with additional delayed-states, the information matrix will exhibit a block tridiagonal structure linking each delayed-state with the post and previous states as shown in (4.6). Hence, the view-based delayed-state SLAM information matrix is *naturally* sparse without having to make any approximations.

$$\begin{bmatrix} \Lambda_{x_{t+1}x_{t+1}} & \Lambda_{x_{t+1}x_t} & & \\ \Lambda_{x_{t+1}x_t}^\top & \Lambda_{x_tx_t} & \Lambda_{x_tx_{t-1}} & \\ & \Lambda_{x_tx_{t-1}}^\top & \Lambda_{x_{t-1}x_{t-1}} & \Lambda_{x_{t-1}x_{t-2}} \\ & & \ddots & \ddots & \ddots \end{bmatrix} \tag{4.6}$$

4.2.2 Measurement Updates

One of the very attractive properties of the information form is that measurement updates are constant-time and additive in an extended information filter (EIF) [160] — this is in contrast to an EKF's quadratic complexity *per* update, which plagued our VAN methodology of Chapter 2. Assume the following general nonlinear measurement function (4.7) and its first-order linearized form (4.8):

$$\begin{aligned} \mathbf{z}_t &= \mathbf{h}(\boldsymbol{\xi}_t) + \mathbf{v}_t & (4.7) \\ &\approx \mathbf{h}(\bar{\boldsymbol{\mu}}_t) + \mathrm{H}(\boldsymbol{\xi}_t - \bar{\boldsymbol{\mu}}_t) + \mathbf{v}_t & (4.8) \end{aligned}$$

where $\boldsymbol{\xi}_t$ is the predicted state vector distributed according to $\boldsymbol{\xi}_t \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_t, \bar{\Sigma}_t) = \mathcal{N}^{-1}(\bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t)$, $\mathbf{v}_t$ is the white measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{R})$, and H is the Jacobian evaluated at $\bar{\boldsymbol{\mu}}_t$. The EKF covariance update requires computing the Kalman gain and updating $\bar{\boldsymbol{\mu}}_t$ and $\bar{\Sigma}_t$ via [7]:

$$\begin{aligned} \mathrm{K} &= \bar{\Sigma}_t \mathrm{H}^\top \big(\mathrm{H}\bar{\Sigma}_t\mathrm{H}^\top + \mathrm{R}\big)^{-1} \\ \boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \mathrm{K}\big(\mathbf{z}_t - \mathbf{h}(\bar{\boldsymbol{\mu}}_t)\big) \\ \Sigma_t &= (\mathrm{I} - \mathrm{KH})\bar{\Sigma}_t(\mathrm{I} - \mathrm{KH})^\top + \mathrm{KRK}^\top. \end{aligned} \tag{4.9}$$

This calculation non-trivially modifies all elements in the covariance matrix resulting in quadratic computational complexity per update [154]. In contrast, the corresponding EIF update is given by [160]:

$$\begin{aligned} \Lambda_t &= \bar{\Lambda}_t + \mathrm{H}^\top\mathrm{R}^{-1}\mathrm{H} \\ \boldsymbol{\eta}_t &= \bar{\boldsymbol{\eta}}_t + \mathrm{H}^\top\mathrm{R}^{-1}\big(\mathbf{z}_t - \mathbf{h}(\bar{\boldsymbol{\mu}}_t) + \mathrm{H}\bar{\boldsymbol{\mu}}_t\big). \end{aligned} \tag{4.10}$$

ESDF Updates are Constant-Time

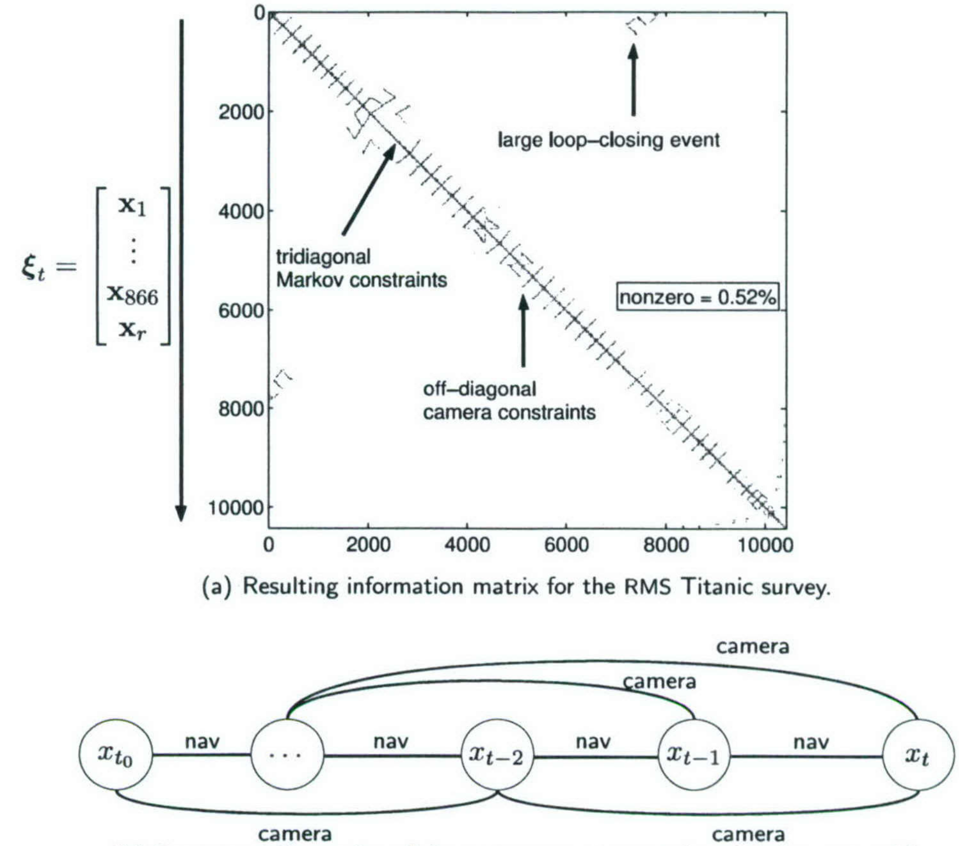
Equation (4.10) shows that the information matrix is additively updated by the outer product term $\mathrm{H}^\top\mathrm{R}^{-1}\mathrm{H}$. In general, this outer product modifies all elements of the predicted information matrix, $\bar{\Lambda}_t$, however, a key observation is that the SLAM Jacobian, H, is always sparse [160]. For example, in the VAN framework of Chapter 2, pairwise registration of images I_i and I_j provides a relative-pose measurement between states $\mathbf{x}_i$ and $\mathbf{x}_j$ resulting in a sparse Jacobian of the form:

$$\mathrm{H} = \begin{bmatrix} 0 \cdots & \frac{\partial \mathbf{h}}{\partial \mathbf{x}_i} & \cdots 0 \cdots & \frac{\partial \mathbf{h}}{\partial \mathbf{x}_j} & \cdots 0 \end{bmatrix}.$$

As a result, only the four block-elements corresponding to $\mathbf{x}_i$ and $\mathbf{x}_j$ of the information matrix need to be modified. In particular, the information in the diagonal blocks $\bar{\Lambda}_{x_i x_i}$ and $\bar{\Lambda}_{x_j x_j}$ is increased, while new information appears at $\bar{\Lambda}_{x_i x_j}$ and its symmetric counterpart $\bar{\Lambda}_{x_j x_i}$. This new off-diagonal information reflects the addition of a new edge into the corresponding Markov network linking the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$.

Putting (4.6) together with (4.10), we see that an important consequence of the delayed-state framework is that the total number of *nonzero* off-diagonal elements in the information matrix is *linear* in the number of state elements and relative-pose constraints (i.e., cam-

Figure 4-1 View-based SLAM is exactly sparse. This figure highlights the exact sparsity of the view-based SLAM information matrix using data from a recent ROV survey of the wreck of the RMS Titanic. In all there are 867 delayed-states where each state is a 12-vector consisting of 6-pose and 6-kinematic components. The resulting information matrix is a 10,404 × 10,404 matrix with only 0.52% nonzero elements.



(a) Resulting information matrix for the RMS Titanic survey.

(b) Conceptual illustration of the measurement constraints (Markov network).

era measurements) for a bounded graph structure. Hence, without any approximation, a view-based representation is exactly sparse and, furthermore, requires only linear storage (Fig. 4-1). In our application, we control the degree of sparisty by bounding the number of image registrations that the robot may attempt per state augmentation. In other words, the robot is only allowed to hypothesize k possible candidate images (where $k = 5$ in our application) for attempted registration with the current view; this leads to at most $2nk$ non-Markov off-diagonal constraints in the resulting information matrix.

As a side note, it is worth pointing out that (4.7) assumes that the measurements are corrupted by time independent noise. Since scan-matching methods rely upon registering raw data, this criteria may be violated if data is reused. In our VAN application, relative-pose measurements are generated by pairwise registration of images with common overlap. As we showed in §2.4.4, this produces motion estimates that are weakly (if at all) correlated for our AUV application. However, for the general case, measurement independence should be ensured by only using a set of raw data correspondences *once*, so that scan-matching measurements remain independent.

4.2.3 Motion Prediction

Motion prediction corresponds to a time propagation of the robot's state from time t to time $t+1$. In (4.5) we derived an expression in the information form for the joint-distribution between the time predicted robot pose, $\mathbf{x}_{t+1}$, and its previous state, $\mathbf{x}_t$ — in other words $p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1})$. To derive the time propagated distribution $p(\mathbf{x}_{t+1}, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1})$, note that all that is required is to simply marginalize out the previous state, $\mathbf{x}_t$, from the joint-distribution in (4.5). Referring to Table 3.1, pg. 87 for marginalization of a Gaussian in the information form we have[2]

$$p(\mathbf{x}_{t+1}, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1}) = \mathcal{N}^{-1}(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}) = \int p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1}) d\mathbf{x}_t$$

$$\begin{aligned}
\bar{\boldsymbol{\eta}}_{t+1} &= \begin{bmatrix} \mathrm{Q}^{-1}\big(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big) \\ \boldsymbol{\eta}_M \end{bmatrix} - \begin{bmatrix} -\mathrm{Q}^{-1}\mathrm{F} \\ \Lambda_{Mx_t} \end{bmatrix} \Omega^{-1}\Big(\boldsymbol{\eta}_{x_t} - \mathrm{F}^\top \mathrm{Q}^{-1}\big(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big)\Big) \\
&= \begin{bmatrix} \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\boldsymbol{\eta}_{x_t} + \Psi\big(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big) \\ \boldsymbol{\eta}_M - \Lambda_{Mx_t}\Omega^{-1}\Big(\boldsymbol{\eta}_{x_t} - \mathrm{F}^\top \mathrm{Q}^{-1}\big(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big)\Big) \end{bmatrix} \\
\bar{\Lambda}_{t+1} &= \begin{bmatrix} \mathrm{Q}^{-1} & 0 \\ 0 & \Lambda_{MM} \end{bmatrix} - \begin{bmatrix} -\mathrm{Q}^{-1}\mathrm{F} \\ \Lambda_{Mx_t} \end{bmatrix} \Omega^{-1} \begin{bmatrix} -\mathrm{F}^\top \mathrm{Q}^{-1} & \Lambda_{x_t M} \end{bmatrix} \\
&= \begin{bmatrix} \Psi & \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\Lambda_{x_t M} \\ \Lambda_{Mx_t}\Omega^{-1}\mathrm{F}^\top \mathrm{Q}^{-1} & \Lambda_{MM} - \Lambda_{Mx_t}\Omega^{-1}\Lambda_{x_t M} \end{bmatrix}
\end{aligned} \tag{4.11}$$

where

$$\Omega = (\Lambda_{x_t x_t} + \mathrm{F}^\top \mathrm{Q}^{-1}\mathrm{F}) \quad \text{and} \quad \begin{aligned} \Psi &= \mathrm{Q}^{-1} - \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\mathrm{F}^\top \mathrm{Q}^{-1} \\ &= \mathrm{Q}^{-1} - \mathrm{Q}^{-1}\mathrm{F}\big(\mathrm{F}^\top \mathrm{Q}^{-1}\mathrm{F} + \Lambda_{x_t x_t}\big)^{-1}\mathrm{F}^\top \mathrm{Q}^{-1} \\ &= (\mathrm{Q} + \mathrm{F}\Lambda_{x_t x_t}^{-1}\mathrm{F}^\top)^{-1}. \end{aligned}$$

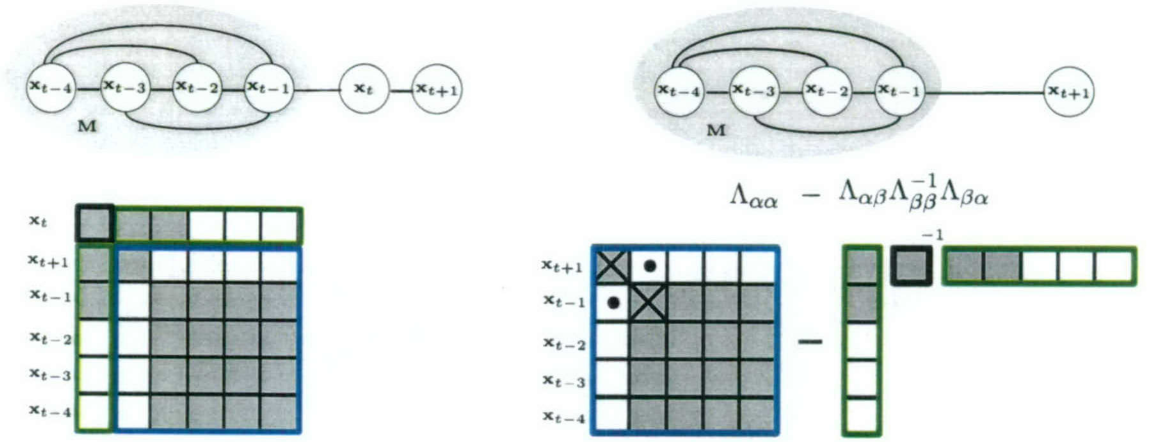
ESDF Prediction is Constant-Time

An important consequence of the delayed-state framework is that (4.11) can be implemented in constant-time. To see this we refer to Fig. 4-2, which illustrates the effect of motion prediction for a collection of delayed-states. We begin with the Markov network of Fig. 4-2(a) showing a segregated collection of delayed-states. Our view-based "map"

[2] The simplification of Ψ employs the matrix inversion lemma:

$$(\mathrm{A} + \mathrm{B}\mathrm{C}\mathrm{B}^\top)^{-1} = \mathrm{A}^{-1} - \mathrm{A}^{-1}\mathrm{B}(\mathrm{B}^\top \mathrm{A}^{-1}\mathrm{B} + \mathrm{C}^{-1})^{-1}\mathrm{B}^\top \mathrm{A}^{-1}.$$

Figure 4-2 ESDF motion prediction is constant-time. Shown below is a graphical illustration of the effect of motion prediction within a delayed-state framework. (a) The Markov network for a segregated collection of delayed-states. The view-based "map", $\mathbf{M}$, is composed of the set $\mathbf{M} = \{\mathbf{x}_{t-4}, \mathbf{x}_{t-3}, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}\}$, which is a collection of delayed-states that are interlinked by camera constraints. The previous and predicted robot states, $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ respectively, are serially linked to the map. Below the Markov network is a schematic showing the nonzero structure (colored in gray) of the associated information matrix. (b) Recalling from Table 3.1 the expression for marginalization of a Gaussian in information form, we see that the bottommost schematic illustrates this operation graphically. The end result is that only the states that were linked to $\mathbf{x}_t$ (i.e., $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t+1}$) are effected by the marginalization operation as indicated by the cross-hairs and black dots superimposed on $\Lambda_{\alpha\alpha}$.



(a) State augmentation of $\mathbf{x}_{t+1}$.

(b) Marginalization over $\mathbf{x}_t$.

corresponds to the set of states $\mathbf{M} = \{\mathbf{x}_{t-4}, \mathbf{x}_{t-3}, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}\}$, which have an interconnected dependence due to camera measurements while the states $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are only serially connected and correspond to the previous and predicted robot states respectively. Referring back to Table 3.1, we see that Fig. 4-2(b) illustrates the effect of marginalization on the information matrix. We note that since $\mathbf{x}_t$ is only serially connected to $\mathbf{x}_{t+1}$ and $\mathbf{x}_{t-1}$, marginalizing it out only requires modifying the information blocks associated with these elements (i.e., $\Lambda'_{x_{t+1}x_{t+1}}$ and $\Lambda'_{x_{t-1}x_{t-1}}$ shown with cross-hairs and the symmetric blocks $\Lambda'_{x_{t+1}x_{t-1}} = \Lambda'^{\top}_{x_{t-1}x_{t+1}}$ shown with black dots). Therefore, since only a fixed portion of the information matrix is ever involved in the calculation of (4.11), motion prediction can be performed in constant-time. This is an important result since in practice the fusion of asynchronous navigation sensor measurements (e.g., odometry, compass) implies that prediction is typically a high-bandwidth operation (e.g., $\mathcal{O}(10\text{ Hz})$ or more).

4.2.4 State Recovery

The information form of the Gaussian is parameterized by its information vector and information matrix, $\boldsymbol{\eta}_t$ and Λ_t respectively. However, the expressions for motion prediction (4.11) and measurement update (4.10) additionally require sub-elements from the state mean vector, $\boldsymbol{\mu}_t$, so that the nonlinear models (4.2) and (4.7) can be linearized. Therefore, in order for the information form to be a computationally efficient parameterization for delayed-states, we also need to be able to easily recover portions of the state mean vector. Fortunately, this is the case due to the sparse structure of the information matrix, Λ_t.

Full State Recovery

Naïve recovery of our state estimate through matrix inversion results in cubic complexity and destroys any efficiency gained over the EKF. Fortunately, closer inspection reveals that the recovery of the state mean, $\boldsymbol{\mu}_t$, can be posed more efficiently as solving the sparse, symmetric, positive-definite, linear system of equations shown in (4.12).

$$\Lambda_t \boldsymbol{\mu}_t = \boldsymbol{\eta}_t \tag{4.12}$$

Such systems can be solved via the classic iterative method of conjugate gradients (CG) [143]. In general, CG can solve this system in n iterations with $\mathcal{O}(n)$ cost per iteration where n is the size of the state vector (i.e., $\mathcal{O}(n^2)$ total cost), and typically in many fewer iterations if the initialization is good [78]. In addition, since the state mean, $\boldsymbol{\mu}_t$, typically does not change significantly with each measurement update (excluding key events like loop-closure), this relaxation can take place over *multiple time steps* using a fixed number of iterations per update as pioneered by Duckett, Marsland, and Shapiro [33] and Thrun et al. [160]. The caveat is that a fixed number of iterations does not necessarily guarantee optimal state recovery [128].

Recently, a couple of newly proposed multilevel relaxation SLAM algorithms that appear capable of solving (4.12) in linear asymptotic complexity (i.e., $\mathcal{O}(n)$) have appeared in the literature. These new state recovery techniques by Konolige [78] and Frese, Larsson, and Duckett [52] achieve the dramatic computational reduction by subsampling poses and performing the relaxation over *multiple spatial resolutions.* Borrowing multigrid relaxation techniques pioneered in the early 1970's for solving discretized partial differential equations (PDEs) [14], the key idea is that spatial subsampling improves relaxation convergence rates. Hence, since measurement updates (§4.2.2) and motion prediction (§4.2.3) are both constant-time operations that depend upon the solution to (4.12), this suggests that ESDFs are at most $\mathcal{O}(n)$ complexity if we employ the relaxation techniques of [52, 78].

Partial State Recovery

An important observation regarding the expressions for motion prediction (4.11) and measurement updates (4.10) is that they only require knowing *subsets* of the state mean $\boldsymbol{\mu}_t$. In light of this we note that rather than solving for the complete state mean vector, $\boldsymbol{\mu}_t$, we can partition (4.12) into two sets of coupled equations as

$$\begin{bmatrix} \Lambda_{\ell\ell} & \Lambda_{\ell b} \\ \Lambda_{b\ell} & \Lambda_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_\ell \\ \boldsymbol{\mu}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_\ell \\ \boldsymbol{\eta}_b \end{bmatrix}. \tag{4.13}$$

This partitioning of $\boldsymbol{\mu}_t$ into what we call the "local portion" of the map, $\boldsymbol{\mu}_\ell$, and the "benign portion", $\boldsymbol{\mu}_b$, allows us to *sub-optimally* solve for the local portion of the map we are interested in constant-time. By holding our current estimate for $\boldsymbol{\mu}_b$ fixed, we can solve

(4.13) for an estimate of $\boldsymbol{\mu}_\ell$ as

$$\hat{\boldsymbol{\mu}}_\ell = \Lambda_{\ell\ell}^{-1}\big(\boldsymbol{\eta}_\ell - \Lambda_{\ell b}\hat{\boldsymbol{\mu}}_b\big). \tag{4.14}$$

Equation (4.14) provides us with a method for recovering an estimate of the local map, $\hat{\boldsymbol{\mu}}_\ell$, provided that our estimate for the benign portion, $\hat{\boldsymbol{\mu}}_b$, is a decent approximation to the actual mean, $\boldsymbol{\mu}_b$. Furthermore, note that only a *subset* of $\hat{\boldsymbol{\mu}}_b$ is actually required in the calculation of $\hat{\boldsymbol{\mu}}_\ell$ corresponding to the nonzero elements in the sparse matrix $\Lambda_{\ell b}$. In terms of Thrun et al.'s notation [160], this active subset, denoted $\boldsymbol{\mu}_b^+$, represents the Markov blanket of $\boldsymbol{\mu}_\ell$ and corresponds to elements that are directly connected to $\boldsymbol{\mu}_\ell$ in the associated Markov network. Therefore, calculation of the local map, $\hat{\boldsymbol{\mu}}_\ell$, only requires an estimate of the *locally* connected delayed-state network, $\hat{\boldsymbol{\mu}}_b^+$, and does not depend upon passive elements in the benign portion of the map.

In particular, (4.14) provides an accurate and constant-time approximation for recovering the robot mean during motion prediction, and during incorporation of high bandwidth navigation sensor measurements. Since the robot state is only serially connected to the map, $\Lambda_{\ell b}$ has only one nonzero block-element (§4.2.3). Therefore, solving for the robot mean is constant-time. Note, though, that (4.14) will only provide a good approximation so long as the active mean estimate, $\hat{\boldsymbol{\mu}}_b^+$, is accurate. In the case that it is not (e.g., as a result of loop closure), then the true full mean, $\boldsymbol{\mu}_t$, should be solved for via (4.12).

4.3 Consistent Covariance Recovery

While recovering the mean is a vital component for making real-world decisions when interacting with the environment, it alone is not always sufficient. For example, robotic tasks such as motion planning, data association, and loop-closing usually require some notion of the joint-uncertainty between the state estimates (i.e., the covariance matrix). Furthermore, estimates of how "certain" we are of map relations can have imperative implications on the action of the robot — quoting Uhlmann [23]:

> An autonomous vehicle controller, for example, might not take evasive action in response to an estimate that places the mean position of the vehicle at the edge of the road and an uncertainty of only one centimeter. But if the same estimate had an uncertainty of a meter, the controller would likely direct the vehicle toward the center of the lane to avoid the worst case possibility that it is actually off the road.

While it is hard to define a single definition of consistency employed uniformly in the prior literature on SLAM, intuitively consistency reflects the goal that the error estimates computed by the filter should "match" the actual errors. In relation to SLAM, consistency of the error estimates is important for data association — determining the correspondences for measurements [118]. This is important both in the context of "local" SLAM (detecting and tracking features), and in a "global" sense (for closing loops). If the SLAM error estimates are too small (over-confident), both of these tasks can become difficult, as will be shown in §4.5.2.

Our strategy for approximate covariance recovery from the information form is formulated upon gaining efficient access to meaningful values of covariance that are consistent with respect to the actual covariances obtained by matrix inversion. The motivation for a consistent approximation is that we guard against under-representing the uncertainty associated with our state estimates, which otherwise could lead to data association and robot planning errors. It is the access to meaningful values of joint-covariance for robot interaction, data association, and decision making in the information form which motivates our discussion.

4.3.1 Setting the Stage

Naïve Covariance Recovery

The covariance matrix corresponds to the inverse of the information matrix, however, actually recovering the covariance via matrix inversion is not practical since this is a cubic operation. Additionally, while the information matrix can be a sparse representation for storage, in general, its inverse results in a *fully dense* covariance matrix despite any sparsity in the information form [51]. This implies that calculating the covariance matrix requires quadratic memory storage — a requirement that could become prohibitive for very large maps (e.g., maps $\geq \mathcal{O}(10^5)$ state elements). To illustrate this point, for the $10,404 \times 10,404$ information matrix shown in Fig. 4-1, storing it in memory only requires 4.5 MB of double precision storage for the nonzero elements while its inverse requires over 865 MB.

What Do We Need?

Fortunately, recovering the full covariance matrix usually isn't necessary for SLAM as many of the data association and robotic planning decisions typically do not require the *entire* covariance matrix, but only the covariance over *subsets* of state variables [30]. Unfortunately, accessing only subsets of state variables in the information form is not an easy task. As we saw in §3.2.2, the covariance and information representations of the Gaussian distribution lead to very different computational characteristics with respect to the fundamental probabilistic operations of marginalization and conditioning.[3] In particular, marginalization is easy in the covariance form since it corresponds to extracting the appropriate sub-block from the covariance matrix while in the information form it is hard because it involves calculating the Schur complement over the variables we wish to keep. Therefore, even though we may only need access to covariances over subsets of the state elements [30] (and thus only have to invert a small information matrix related to the subset of variables we are interested in), accessing them in the information form requires marginalizing out most of our state vector resulting in cubic complexity due to matrix inversion in the Schur complement.

Prior Art

To get around this dilemma for SEIFs, Thrun et al. [92, 160] proposed a data association strategy based upon using *conditional* covariances. Since conditional information matrices are easy to obtain in the information form (simply extract a sub-block over the desired

[3] See Table 3.1, pg. 87.

variables), their strategy is to choose an appropriate sub-block from the information matrix such that it's inverse *approximates* the actual covariance for the subset of variables they are interested in. In particular, given two state variables of interest, $\mathbf{x}_i$ and $\mathbf{x}_j$, their approximation selects the joint Markov blanket, $\mathbf{M}_i^+ \cup \mathbf{M}_j^+$ (i.e., $\mathbf{M}_k^+$ represents state variables *directly* connected to $\mathbf{x}_k$ in a graph theoretic sense within the information matrix), and additionally if the intersection is null (i.e., $\mathbf{M}_i^+ \cap \mathbf{M}_j^+ = \emptyset$), variables along a path connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ topologically. Their method then extracts and inverts this sub-block to obtain a joint-covariance matrix for $\mathbf{x}_i$ and $\mathbf{x}_j$ conditioned on all other variables that have an indirect influence. They note that empirical testing shows that their approximation method seems to work well in practice for their application [92], despite the fact that using conditional covariances should result in an *over-confident* approximation.

4.3.2 Consistent Covariances for Data Association

In this section we outline our strategy for recovering approximate joint-covariances useful for *data association.* Before we begin, we want it to be clear to the reader that our technique for obtaining and maintaining these covariances should not be confused with the actual updating and mechanics of the information parameterization. What we present in the following is a way of maintaining covariance *bounds* that are consistent with respect to the information parameterization. Furthermore, these covariances are used for data association *only* and are not in any way involved in the actual update and maintenance of the information filter representation. With that being said we now present our algorithm.

Efficiently Accessing the Robot's Covariance

We begin by noting that recovery of our state estimate, $\boldsymbol{\mu}_t$, from the information form already requires that we solve the sparse, symmetric, positive-definite system of equations (4.12) and moreover that this system can be solved in $\mathcal{O}(n)$ time using [52, 78]. Our covariance recovery strategy for the information form is based upon augmenting this linear system of equations so that the current robot pose covariance is accessible as well. Note that by definition $\Lambda_t \Sigma_t = \mathrm{I}$ and, therefore, by picking the i^{th} basis vector $\mathbf{e}_i$ from the identity matrix we can use it to selectively solve for a column of the covariance matrix, denoted as Σ_{*i}.

$$\Lambda_t \Sigma_t = \mathrm{I} \quad \Rightarrow \quad \Lambda_t \Sigma_{*i} = \mathbf{e}_i \quad \text{where} \quad \mathrm{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$$

To obtain the robot's covariance at any time step we simply augment our original linear system (4.12) to include an appropriate set of basis vectors $\mathrm{E}_r = \{\mathbf{e}_r\}$ such that the solution to (4.15) provides access to our current state and the robot's covariance-column.

$$\Lambda_t \begin{bmatrix} \boldsymbol{\mu}_t & \Sigma_{*r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_t & \mathrm{E}_r \end{bmatrix} \tag{4.15}$$

Inserting a New Map Element

Given that (4.15) provides a mechanism for efficient access to the robot's covariance-column, Σ_{*r}, we exploit it to obtain useful covariance bounds for other map elements. For example, whenever we insert a new image I_i into our view-based map, we must correspondingly add a new element $\mathbf{x}_i$ into our view-based SLAM state vector [36, 38]. This new state element corresponds to a sampling of our robot state at time t_i (i.e., $\mathbf{x}_i = \mathbf{x}_r(t_i)$) and represents our estimate of where the robot was when it took that image. Since the two states are coincident at time t_i, the covariance for $\mathbf{x}_i$ is $\Sigma_{ii} = \Sigma_{rr}$ and can be obtained by solving (4.15). A well-known property of SLAM is that over time the covariance for $\mathbf{x}_i$ will *decrease* as new sensor measurements are incorporated and all map elements become fully correlated [30]. Therefore, storing $\tilde{\Sigma}_{ii} = \Sigma_{ii}$ as our initial approximate covariance estimate for $\mathbf{x}_i$ serves as a *conservative* bound to the actual marginal covariance for all time, (i.e., $\tilde{\Sigma}_{ii} \geq \Sigma_{ii}(t)$).

Data Association

In our application, the joint-covariance between the time-projected robot pose, $\mathbf{x}_r$, and any other map entry, $\mathbf{x}_i$, i.e.,

$$\bar{\Sigma}_{joint} = \begin{bmatrix} \bar{\Sigma}_{rr} & \bar{\Sigma}_{ri} \\ \bar{\Sigma}_{ri} & \Sigma_{ii} \end{bmatrix}$$

is needed for two operations: link proposal (§2.3.3) and pose-constrained correspondence searches (§2.4.3). Link proposal corresponds to hypothesizing which images in our view-based map could potentially share common overlap with the current image being viewed by the robot, denoted I_r, and, therefore, could potentially be registered to generate a relative-pose measurement. The second operation, pose-constrained correspondence searches, uses the relative-pose estimate between candidate images I_i and I_r to restrict the image-based correspondence search to probable regions based upon a two-view point transfer relation.[4]

To obtain the actual joint-covariance, $\bar{\Sigma}_{joint}$, from the information form requires marginalizing out all other elements in our map except for $\mathbf{x}_r$ and $\mathbf{x}_i$ and leads to cubic complexity in the number of eliminated variables. However, we can obtain a bounded approximation to $\bar{\Sigma}_{joint}$ at any time-step by using the solution from (4.15) to provide us with the current covariance-column, $\bar{\Sigma}_{*r}$, representing the joint-covariances between the time-projected robot and all other map entries (note that this solution is equivalent to what could be obtained by full matrix inversion of $\bar{\Lambda}_t$). Using this result we can construct a conservative joint-covariance approximation to $\bar{\Sigma}_{joint}$ as

$$\tilde{\bar{\Sigma}}_{joint} = \begin{bmatrix} \bar{\Sigma}_{rr} & \bar{\Sigma}_{ir}^{\top} \\ \bar{\Sigma}_{ir} & \tilde{\Sigma}_{ii} \end{bmatrix} \tag{4.16}$$

where $\bar{\Sigma}_{rr}$ and $\bar{\Sigma}_{ir}$ are extracted from $\bar{\Sigma}_{*r}$, and $\tilde{\Sigma}_{ii}$ is our conservative covariance bound for $\mathbf{x}_i$. Note that (4.16) represents a valid positive-semidefinite and, therefore, consistent approximation satisfying

$$\tilde{\bar{\Sigma}}_{joint} - \bar{\Sigma}_{joint} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\Sigma}_{ii} - \Sigma_{ii} \end{bmatrix} \geq 0$$

since $\tilde{\Sigma}_{ii} - \Sigma_{ii} \geq 0$. Given that (4.16) provides a consistent approximation to the true covariance, we can use it to compute conservative first-order probabilities of relative-pose (i.e., $\mathbf{x}_{ri} = \ominus\mathbf{x}_r \oplus \mathbf{x}_i$) for link hypothesis and correspondence searches.

[4] Note that the standard maximum likelihood data association technique for feature-based SLAM also only depends on extracting $\bar{\Sigma}_{joint}$ [30].

Updating the Covariance Bounds

Since $\tilde{\Sigma}_{ii}$ serves as a *conservative* approximation to the actual covariance, Σ_{ii}, for map element $\mathbf{x}_i$, we would like to be able to place tighter bounds on it as we gather more measurement information. In fact, the careful reader will recognize that our SLAM information filter *is implicitly already doing this* for us, however the issue is that extracting the actual filter bound, Σ_{ii}, from the information matrix representation is not particularly convenient. Note that while we could access Σ_{ii} by solving for the covariance-column Σ_{*i} using an appropriately chosen set of basis vectors, the reason for not doing this is that iteratively solving systems like (4.15) is efficient only when we have a good starting point [33, 78]. In other words, when we solve (4.15) for the latest state and robot covariance-column, our estimates $\boldsymbol{\mu}_t$ and Σ_{*r} from that last time-step serve as good seed points and, therefore, typically only require a small number of iterations per time-step to update (excluding loop-closing events). In the case of solving for an arbitrary column, Σ_{*i}, we do not have a good *a priori* starting point and thus convergence will be slower.

Our approach for tightening the bound $\tilde{\Sigma}_{ii}$ is to use our joint-covariance approximation (4.16) and perform a simple constant-time Kalman update on a *per* re-observation basis. In other words, we only update our covariance bound, $\tilde{\Sigma}_{ii}$, when the robot re-observes $\mathbf{x}_i$ and successfully generates a relative-pose measurement, $\mathbf{z}_{ri}$, by registering images I_i and I_r. We then use that relative-pose measurement to perform an EKF update (4.9) on the *fixed size* state vector $\mathbf{y} = \left[\mathbf{x}_r^\top, \mathbf{x}_i^\top\right]^\top$ obtaining the new conservative bound $\tilde{\Sigma}_{ii}^{+}$.

Mathematically, the distribution over $\mathbf{y}$ corresponds to marginalizing out all elements in our state vector except for $\mathbf{x}_r$ and $\mathbf{x}_i$ as

$$p(\mathbf{y}) = \int_{\mathbf{x}_j \neq \{\mathbf{x}_r, \mathbf{x}_i\}} \mathcal{N}^{-1}(\bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t) d\mathbf{x}_j = \int_{\mathbf{x}_j \neq \{\mathbf{x}_r, \mathbf{x}_i\}} \mathcal{N}(\bar{\boldsymbol{\mu}}_t, \bar{\Sigma}_t) d\mathbf{x}_j, \tag{4.17}$$

which results in the distribution:

$$p(\mathbf{y}) = \mathcal{N}\Big(\begin{bmatrix}\bar{\boldsymbol{\mu}}_r \\ \bar{\boldsymbol{\mu}}_i\end{bmatrix}, \begin{bmatrix}\bar{\Sigma}_{rr} & \bar{\Sigma}_{ir}^\top \\ \bar{\Sigma}_{ir} & \Sigma_{ii}\end{bmatrix}\Big). \tag{4.18}$$

Recalling that (4.16) already provides us with a consistent approximation to this distribution, we have

$$\tilde{p}(\mathbf{y}) = \mathcal{N}\Big(\begin{bmatrix}\bar{\boldsymbol{\mu}}_r \\ \bar{\boldsymbol{\mu}}_i\end{bmatrix}, \begin{bmatrix}\bar{\Sigma}_{rr} & \bar{\Sigma}_{ir}^\top \\ \bar{\Sigma}_{ir} & \tilde{\Sigma}_{ii}\end{bmatrix}\Big) \tag{4.19}$$

where the only difference between the actual distribution (4.18) and its approximation (4.19) is the conservative marginal $\tilde{\Sigma}_{ii}$. Using the measurement $\mathbf{z}_{ri}$ we now perform a Kalman update (4.9) on (4.19) yielding the conditional distribution $\tilde{p}(\mathbf{y}|\mathbf{z}_{ri})$ from which we retain only the updated marginal bound $\tilde{\Sigma}_{ii}^{+}$ for element $\mathbf{x}_i$. This update is computed in constant-time for each re-observed feature. Algorithm 4.1 summarizes the procedure.

Note that by conceptually performing the marginalization step of (4.17) before computing the Kalman update, we have avoided any inconsistency issues associated with only storing the marginal bounds, $\tilde{\Sigma}_{ii}$, and not representing the intra-map correlations. This ensures that our update step will result in a consistent marginal bound for data association that will improve over time as we re-observe map elements.

Algorithm 4.1 Calculation of marginal covariance bounds used for data association.

```
Require: $\Sigma_{*r}$ {initialize bound}
if $\mathbf{x}_i$ = new map element then
    store $\tilde{\Sigma}_{ii} \leftarrow \Sigma_{rr}$
end if

Require: $\bar{\mu}_t, \bar{\Sigma}_{*r}$ {data association and bound update}
for all $\mathbf{x}_i$ do
    $\tilde{\bar{\Sigma}}_{joint} \leftarrow \begin{bmatrix} \bar{\Sigma}_{rr} & \bar{\Sigma}_{ri} \\ \bar{\Sigma}_{ri} & \tilde{\Sigma}_{ii} \end{bmatrix}$
    compute link hypothesis as outlined in §2.3.3
    if candidate link then
        do constrained correspondence search on $I_i$ and $I_r$ as outlined in §2.4.3
        if image registration success then
            do Kalman update on $\tilde{\bar{\Sigma}}_{joint}$ using measurement $\mathbf{z}_{ri}$
            store $\tilde{\Sigma}_{ii} \leftarrow \tilde{\Sigma}_{ii}^{+}$
        end if
    end if
end for
```

4.4 Discussion

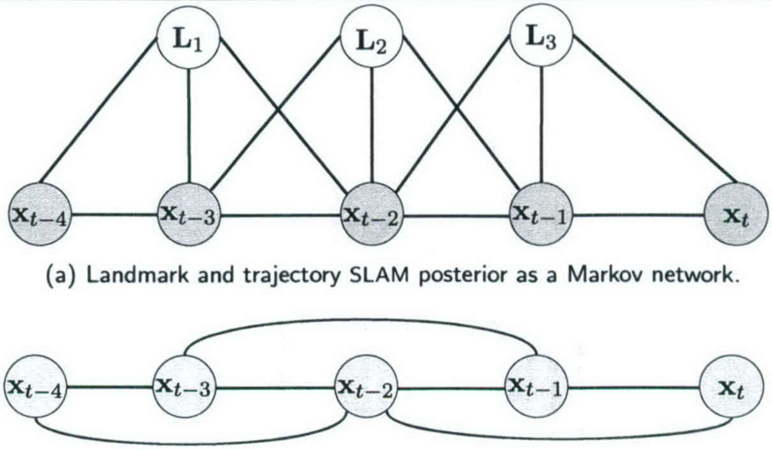
4.4.1 Connection to Lu-Milios

The concept of a view-based map representation has strong roots going back to a seminal paper by Lu and Milios [94]. Their approach sidestepped difficulties associated with feature segmentation and representation by doing away with an explicit feature-based parameterization of the environment. Rather, their technique indirectly represented a physical map via a collection of global robot poses and raw scan data. To determine the global poses, they formulated the nonlinear optimization problem as one of estimating a set of global robot poses consistent with the relative-pose constraints obtained by scan matching and odometry. They then solved this sparse nonlinear optimization problem in an batch-iterative fashion. Our ESDF framework essentially attempts to recursively solve the same problem. Note, though, that in the ESDF framework the nonlinear relative-pose constraints are only linearized *once* about the current state when the measurement is incorporated via (4.10) while in the noncausal Lu-Milios batch formulation they are *re-linearized* around the current best estimate of the state at *each* iteration of the nonlinear optimization. This implies that while the ESDF solution can be performed recursively, it will be more prone to linearization error.

4.4.2 Connection to Feature-Based SLAM

Another interesting theoretical connection involves relating the delayed-state SLAM framework to feature-based SLAM. In Chapter 3 we saw that the feature-based SLAM information matrix is naturally dense as a result of marginalizing out the robot's trajectory. On a sim-

Figure 4-3 ESDF's connection to feature-based SLAM. Conceptually, view-based SLAM can be viewed as marginalizing out the landmarks (i.e., $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3$), which in turn causes edges to appear between spatially-local samples from the robot's trajectory.



(a) Landmark and trajectory SLAM posterior as a Markov network.

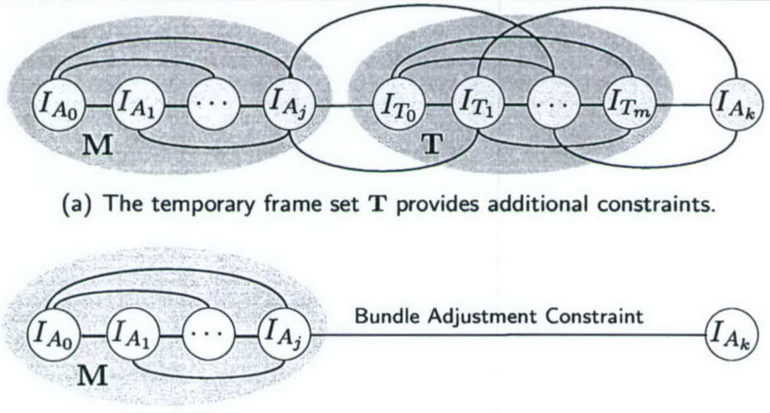
(b) Delayed-state Markov network after marginalizing out the landmarks.

ilar train-of-thought, conceptually we can view the off-diagonal elements appearing in the delayed-state SLAM information matrix as being a result of marginalizing out the *landmarks* as illustrated by Fig. 4-3. Since landmarks are only ever locally observed, they only create links to spatially close robot states. Therefore, each time we eliminate a landmark, it introduces a new off-diagonal entry into the information matrix that links all robot states that observed that landmark. Interestingly, this same type of constraint phenomenon also appears in photogrammetry and in particular in large scale bundle adjustment techniques [163]. These techniques are based upon a partitioned Levenberg-Marquardt algorithm that takes advantage of the inherent sparsity between camera and 3D feature constraints in the reconstruction problem. Their central component is based upon eliminating 3D-structure equations to yield a coupled set of equations over camera poses that they solve and then back-substitute to recover the associated 3D-structure. Therefore, loosely speaking, the delayed-state information framework represents a recursive linearized formulation of this same problem.

4.4.3 Video Frame Rates

Finally, note that a view-based representation is still applicable even with much higher frame rates. In our underwater VAN application, a digital-still image is collected every few seconds from a down-looking monocular camera. Since this typically results in temporal overlap of the order of 15–35%, we include all image frames into our view-based map representation. However, in the general case where much higher frame rates are available, note that we can still use a view-based methodology by selectively *subsampling* key frames from our video sequence to serve as spatial "anchor points" in our view-based map. Reobservation of these key frames (coupled with successful image registration) provides a zero-drift spatial measurement of robot motion allowing us to "close-the-loop". Furthermore, we can exploit the higher frame rates to get an improved estimate of visual odometry

Figure 4-4 View-based SLAM with video frame rates. (a) Our collection of anchor images $\{I_{A_0}, \ldots, I_{A_j}\}$ represents a *subsampling* of the available video image sequence and serves as our view-based spatial map $\mathbf{M}$. Given higher-frame rates, though, we can exploit the additional views between temporally consecutive anchor images I_{A_j} and I_{A_k} to get an improved estimate of incremental motion. The improved motion estimate comes from a local bundle adjustment that includes the temporary frame set $\mathbf{T} = \{I_{T_0}, \ldots, I_{T_m}\}$. (b) The result is a serial constraint between I_{A_j} and I_{A_k} that is much more robust than a single pairwise measurement between the pair.



(a) The temporary frame set $\mathbf{T}$ provides additional constraints.

(b) The result is an improved visual-odometry constraint.

by performing a local-bundle adjustment over all frames occurring between temporally consecutive anchor images (Fig. 4-4). Similar in spirit to Zhang and Shan's visual odometry [181], the local-bundle adjustment provides an improved estimate of incremental camera motion because it incorporates additional constraints. However, by also maintaining a collection of anchor images in our framework, we retain the ability to close-the-loop (with order $\mathcal{O}(n)$ complexity).

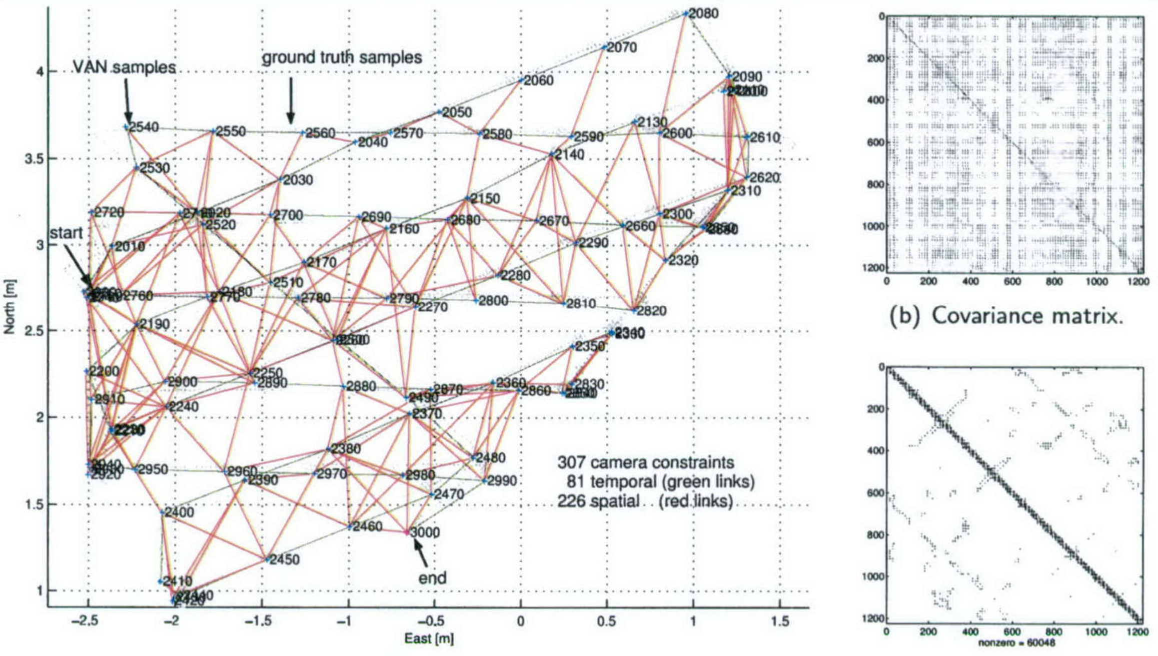
4.5 Results

4.5.1 Experimental Validation: VAN EKF vs. VAN ESDF

Experimental Setup

In this section we demonstrate the efficiency of the ESDF information framework as compared to the EKF-based formulation of Chapter 2. For this purpose, we use the JHU dataset from §2.5.2. Note that all processing was done using MATLAB R13 running on an Intel 1.8 GHz Pentium-4M laptop with 1024 MB of RAM. For the purposes of benchmark comparison, we employed the full state recovery technique of (4.12) after every camera measurement and otherwise used the constant-time partial state recovery method of (4.13) to advance the robot state. To briefly recap, the experimental setup consisted of a downward-looking digital-still camera mounted on a moving underwater pose instrumented ROV at the JHU Hydrodynamic Test Facility. Their vehicle is instrumented with a typical suite of oceanographic dead-reckoning navigation sensors capable of measuring heading, attitude, XYZ bottom-referenced Doppler velocities, and a pressure sensor for depth.

Figure 4-5 This figure contrasts the exact sparsity of the ESDF information matrix versus the density of the full covariance matrix. (a) Spatial topology of a 101 image sequence of underwater images collected from the JHU ROV — in all there are 307 camera constraints. (b) Nonzero elements of the covariance matrix; all elements above a normalized correlation score of 10% are shown. (c) Nonzero elements of the information matrix. Note that the covariance matrix has $1224^2 = 1,498,176$ nonzero elements while the information matrix has only 60,048. The covariance matrix and information matrix are numerically equivalent, however the information matrix is exactly sparse.



(a) VAN recovered trajectory for 101 images of the JHU dataset.

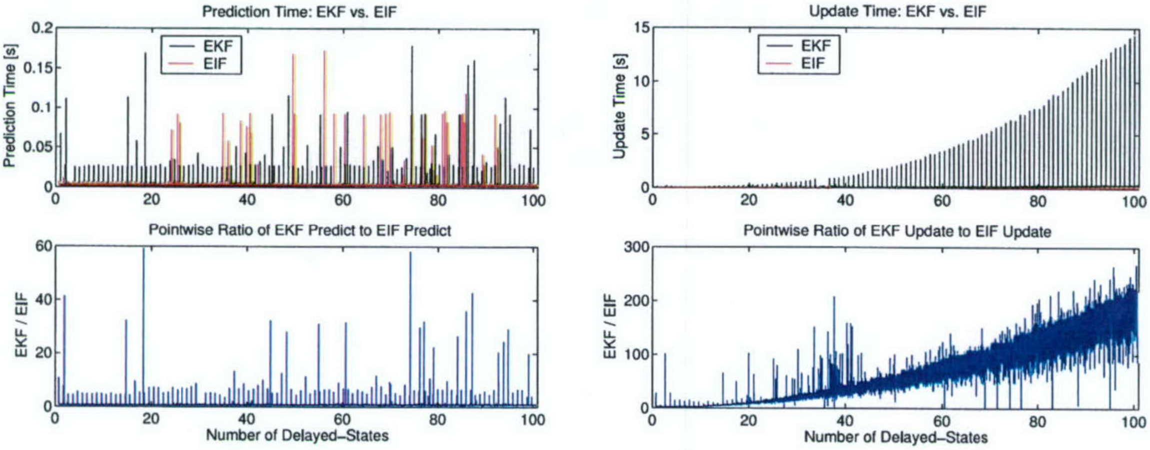
(b) Covariance matrix.

(c) Information matrix.

Experimental Results

Fig. 4-5 shows the result of estimating the ROV delayed-states associated with a 101 image sequence using a full covariance EKF and sparse EIF. For this experiment, the vehicle started near the top-left corner of the plot at (-2.5,2.75) and then drove a course consisting of two grid-based surveys, one oriented SW to NE and the other W to E. The top plot shows the spatial XY pose topology, 3-sigma confidence bounds, and link network of camera constraints. Green links correspond to temporally consecutive images that were successfully registered while red links correspond to spatially registered image pairs. The rightmost plot in this figure compares the densities associated with the EKF covariance matrix versus the ESDF information matrix. Note that while the EKF correlation matrix is dense, the information matrix exhibits a sparse tridiagonal structure with the number of off-diagonal elements being linear in the number of camera measurements. In all there are 307 camera constraints (81 temporal / 226-spatial) and each delayed-state is a 12-vector consisting of 6-pose and 6-kinematic components. Therefore, 102 delayed-states (101 images plus the robot) results in a 1224×1224 information matrix containing $12^2(102 + 2 \cdot 101) + 6^2(2 \cdot 226) = 60,048$ nonzero elements as shown. We found the EKF and ESDF solutions to be numerically equivalent, and furthermore, that the ESDF only required

Figure 4-6 Time comparison of EKF vs. ESDF filtering operations using the JHU dataset. (a) The top figure shows both the EKF and ESDF prediction times in seconds versus the number of delayed-state entries while the bottom figure shows their pointwise ratio. From the plots we can gather that prediction is a constant-time operation for both filters. (b) Same plot layout as before, but now we show the update times for each filter. For benchmark comparison we employed the full-state recovery technique of (4.12) after every camera measurement (using MATLAB's "left divide" capability). Note that despite this, the bottom figure shows that as the number of delayed-state elements increases, the ESDF becomes more efficient relative to the EKF due to the decreasing density of the information matrix.



(a) EKF vs. ESDF prediction times.

(b) EKF vs. ESDF update times.

4% of the storage of the EKF for this experiment.

Turning our attention now to filter efficiency, in Fig. 4-6 we compare the prediction and update times of the EKF to those of the ESDF. In particular, we see that prediction is essentially a constant-time operation for both filters. However, Fig. 4-6(b) shows that ESDF updates are orders of magnitude more efficient than corresponding EKF updates, and moreover that they become *more* efficient relative to the EKF as the number of delayed-states *increases*. This increase in relative efficiency with state size results from a *decreasing* density in the information matrix. Also, note that this impressive computational reduction is despite the fact that we are using MATLAB's "left-divide" capability to solve (4.12) (essentially a form of Gaussian elimination). Hence, the ESDF's results could be even *better* if we implemented the iterative multilevel state recovery techniques of [52,78]. In summary, for this 101 image sequence: data collection took a total of 17 minutes, EKF processing required 29 minutes, and ESDF estimation was just over a minute (i.e., 17× faster than real-time).[5]

[5] Excludes image registration time.

Figure 4-7 Institute for Exploration ROV Hercules and its sensor characteristics [24].

(a) ROV Hercules.

Measurement	**Sensor**	**Precision**
Roll/Pitch	Tilt Sensor	±0.1°
Heading	North-Seeking FOG	±0.1°
Body Frame Velocities	Acoustic Doppler	±0.01 m/s
Depth	Pressure Sensor	±0.01 m
Altitude	Acoustic Altimeter	±0.1 m
Down-looking Imagery	Calibrated 12-bit CCD	1 frame every 8 s

(b) Hercules' pose sensor characteristics.

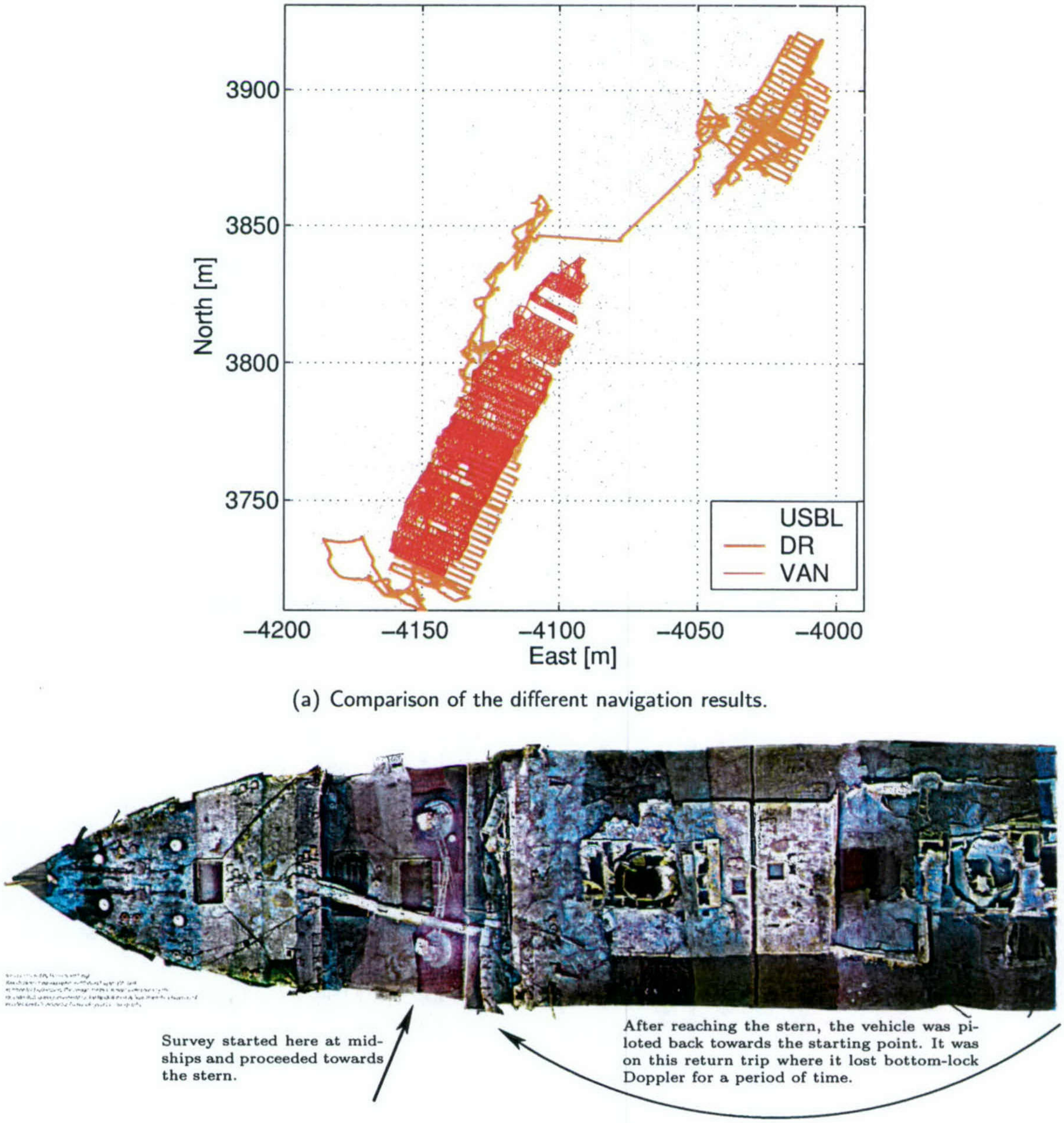
4.5.2 Real-World Results: RMS Titanic

Experimental Setup

This section presents experimental results validating both the large-area scalability of our ESDF framework and the consistency of our covariance recovery technique. The wreck of the RMS Titanic was surveyed during the summer of 2004 by the deep-sea ROV Hercules [24] (Fig. 4-7) operated by the Institute for Exploration of the Mystic Aquarium. The ROV was equipped with a standard suite of oceanographic dead-reckon navigation sensors capable of measuring heading, attitude, altitude, XYZ bottom-referenced Doppler velocities, and a pressure sensor for depth. In addition, the vehicle was also equipped with a calibrated stereo rig consisting of two downward-looking 12-bit digital-still cameras that collected imagery at a rate of 1 frame every 8 seconds. However, the results being presented here were produced using imagery from *one* camera only — the purpose of this self-imposed restriction to a monocular sequence is to demonstrate the general applicability of our VAN methodology.

Fig. 4-8 provides a summary of the survey pattern and comparison of the different navigation methods used for localizing the vehicle. For real-time control, the vehicle integrated bottom-lock Doppler velocity measurements to get a dead-reckoned estimate of XY position. Additionally, ship-based ultra-short-baseline (USBL) tracking provided range and bearing fixes to the vehicle. Since the wreck lies at a depth of approximately 3750 m, the large ship-to-vehicle moment arm, coupled with angular error in the USBL bearing measurements, resulted in an almost useless measurement of vehicle tracking as indicated by the widely

Figure 4-8 Mapping results from a summer of 2004 ROV survey of the RMS Titanic. (a) XY plot comparing the raw dead-reckon navigation data (brown), ship-board ultra-short baseline tracking (gray), and reconstructed VAN trajectory (red). (b) A photomosaic of the RMS Titanic constructed from over 700 digital still images. Note that this photomosaic is presented for visualization purposes only as a representation of the data that serves as input to our algorithm. It is the result of semi-automatic processing with manual selection of a number of common scene points to guide the photomosaicking process. This could be considered as a form of benchmark against which fully autonomous processing can be compared.



(a) Comparison of the different navigation results.

Survey started here at midships and proceeded towards the stern.

After reaching the stern, the vehicle was piloted back towards the starting point. It was on this return trip where it lost bottom-lock Doppler for a period of time.

(b) Photomosaic of the RMS Titanic (courtesy Hanumant Singh, WHOI).

distributed scatter of fixes in Fig. 4-8(a).

The vehicle survey consisted of a grid-based trajectory containing both temporal and side-to-side cross-track overlap. The survey started at midships and proceeded towards the stern. Upon reaching the aft portion of the wreck, the camera was turned off and the vehicle was piloted back towards the starting point. During its return trip, the vehicle lost bottom-lock Doppler velocity measurements for a period of time, and therefore, was unable to dead-reckon integrate its vehicle position during this time period — this is the cause of the "discontinuity" in the brown trajectory of Fig. 4-8(a). After the vehicle returned near its starting point, the camera was turned back on and the vehicle finished off the survey by mapping the bow portion of the wreck.

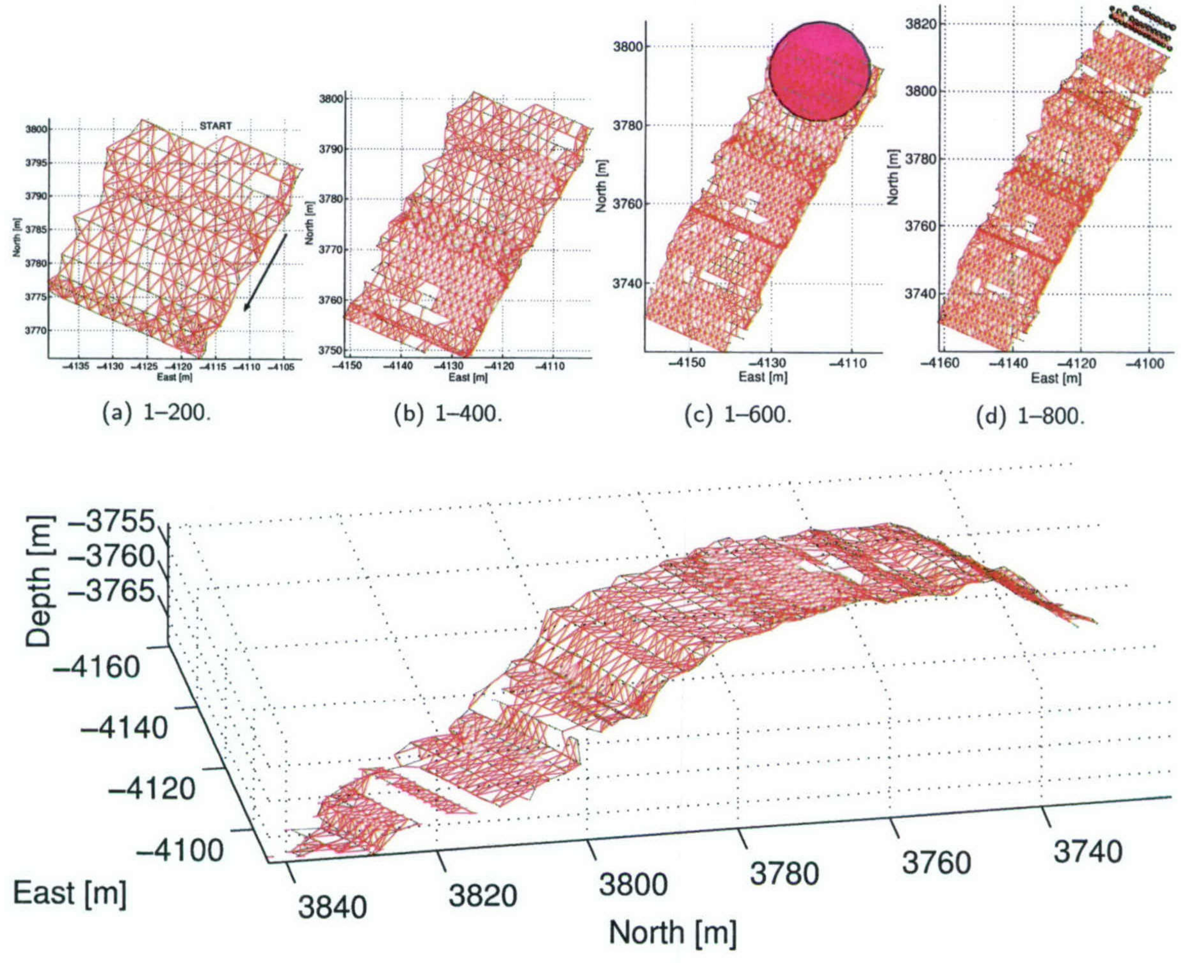
Experimental Results: Scalability

Fig. 4-8(a) shows our post-processed VAN trajectory overlaid in red on top of the traditional oceanographic navigation results. Note that this result was computed from camera and dead-reckon sensors only and does not in anyway use the USBL ship-based tracking. This result constitutes a significant advancement in the current state-of-the-art as it represents the largest visually navigated underwater dataset to date. In support of this claim, note that the vehicle traversed a 2D path length of 3.1 km, and a 3D XYZ path length of 3.4 km during its maneuvers to maintain a safe altitude between it and the wreck. The convex hull of the final mapped region encompasses an area over 3100 m^2 and in all a total of 866 images were used to provide 3494 camera-generated relative-pose constraints.

In Fig. 4-9(a)–(d) we see a time progression of the camera constraints and vehicle trajectory estimate. In particular, Fig. 4-9(c) shows a large loop-closing event where the vehicle successfully re-localized by correctly registering 4 image pairs out of 64 hypothesized candidates after having lost bottom-lock Doppler velocity measurements for an extended period of time. Fig. 4-9(e) shows the final 3D pose-constraint network and Fig. 4-10 shows its 2D view. While there is no ground-truth for this dataset, the resulting pose-network qualitatively appears to be consistent in that the recovered vehicle trajectory forms the outline of a ship's hull. To quantitatively corroborate this observation we pairwise triangulated scene structure using only our image-to-image correspondences and VAN estimated vehicle poses, the results are shown in Fig. 4-11 and Fig. 4-12. In particular, a striking feature of both figures is that the triangulated scene structure exhibits good coherency both globally and locally. This result is even more impressive when taking into consideration the fact that VAN does not explicitly enforce consistency of structure, instead only consistency of poses. This furthermore adds evidence that VAN's global pose estimates are near ideal. As an aside, note that the quality of VAN's results suggests that it can be used as a recursive scalable solution to large-area structure-from-motion since the estimated pose and triangulated structure should provide a good initialization point in an optimal bundle-adjustment step.

Finally, Fig. 4-13 provides a histogram of VAN's precision, expressed as a percentage of distance traveled, for the resulting pose-network. For comparison purposes, figures-of-merit in the literature for state-of-the-art dead-reckoned DVL position error are 1% for a typical heading reference [16] and 0.1% or better when combined with an INS and FOG [101]. Note that VAN's (DVL, plus FOG, plus camera) filter estimated precision is (on a majority) better than 0.005% distance traveled for this dataset (note that 0.005% over 3.1 km equals

Figure 4-9 Time-evolution of the RMS Titanic pose constraint network. (a)–(d) Time progression with 3-sigma bounds, from left to right: images 1–200, 1–400, 1–600, 1–800. Green links represent temporally consecutive registered image pairs while red links represent spatially registered image pairs. Note the large loop-closing event that occurs in (c) when the vehicle returns to the bow of the ship after having traveled from the stern with the camera turned off. (e) XYZ view of the final 3D pose-constraint network associated with using 866 images to provide 3494 camera constraints; 3-sigma bounds are unviewable at this scale. Note that the recovered vehicle poses and image correspondences can be used as direct inputs into a standard bundle adjustment algorithm for structure recovery.



(a) 1–200.

(b) 1–400.

(c) 1–600.

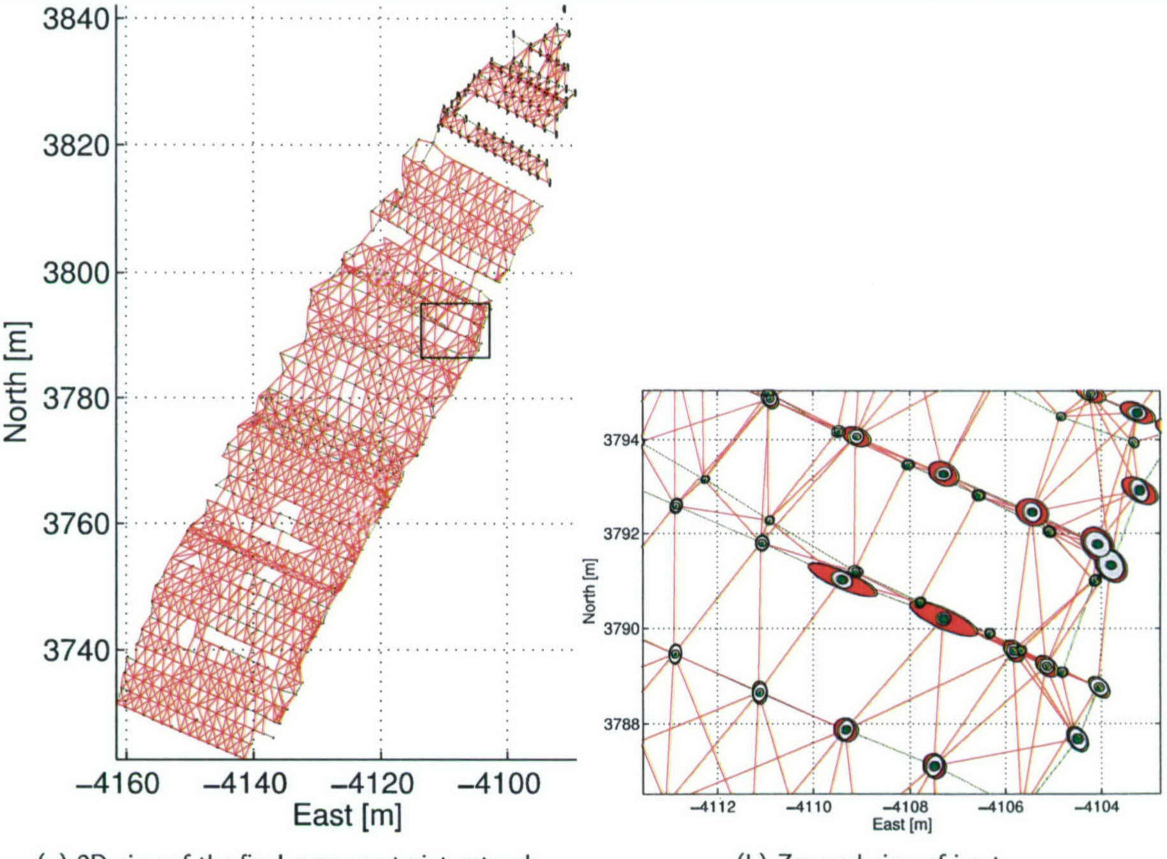
(d) 1–800.

(e) Final 3D pose constraint network.

Survey Summary

- 866 digital-still images
- 3494 camera constraints
- Path length: 2D 3.1 km / 3D 3.4 km
- Convex hull of the mapped area > 3100 m^2

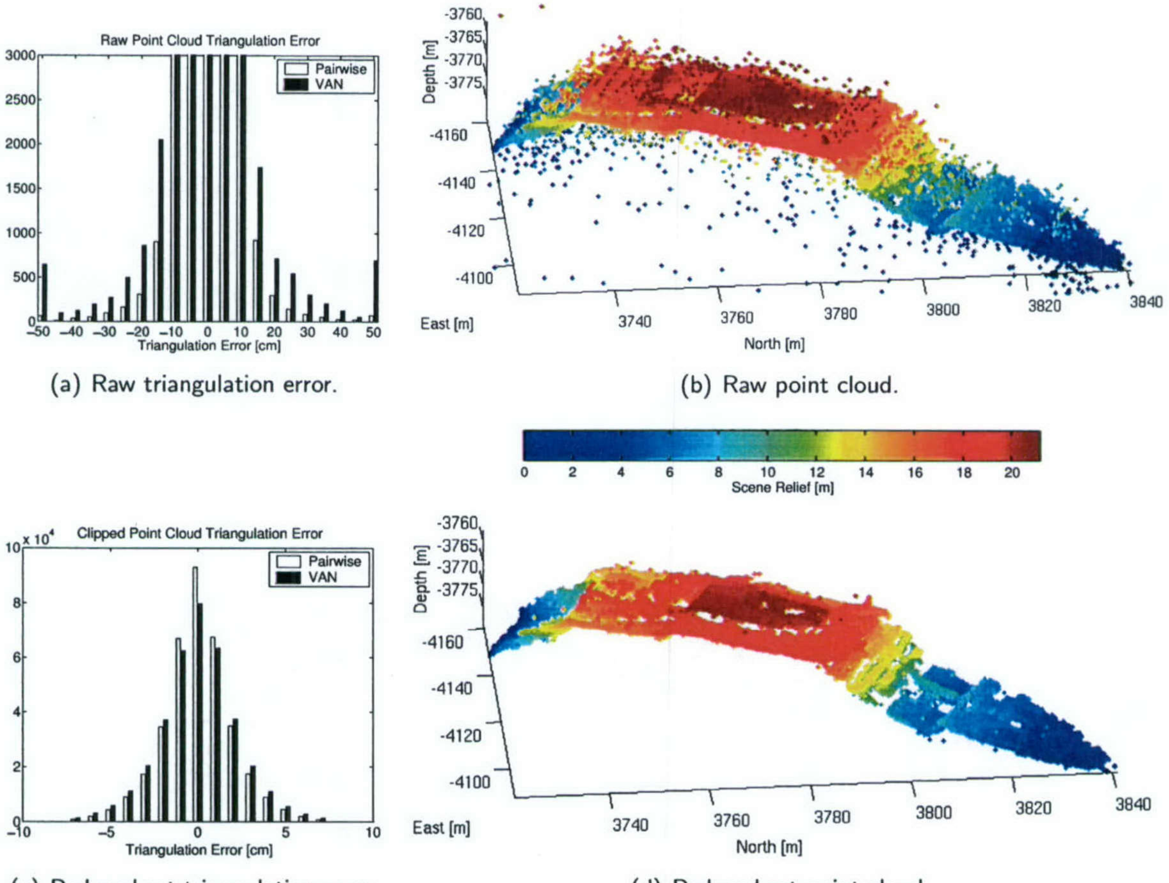
Figure 4-10 The recovered XY trajectory and covariance bounds for the RMS Titanic. (a) Final XY pose constraint-network associated with using 866 images to provide 3494 camera constraints; 3-sigma bounds are shown. (b) A zoomed view illustrating the consistency of the data association bounds generated by our algorithm. Note that for this plot the 3-sigma bounds have been inflated by a factor of 30 for visualization. In this plot we have: 1) the initial covariance bounds associated with pose insertion (red), 2) the marginal covariance bounds based upon constant-time Kalman updates (gray), and 3) the actual marginal covariance bounds obtained by matrix inversion (green). Note that all of the estimated bounds (gray) were verified to be consistent with respect to the actual marginal covariance (green) by performing Cholesky decomposition on their difference to establish positive definiteness. The reason why some of the estimated covariance bounds (gray) are tighter approximations than others to the actual filter bounds (green) is because our algorithm only updates the bounds on a per re-observation basis. Hence, if the robot is sufficiently well-localized when it re-observes an image, then the covariance bound of the corresponding delayed-state improves.



(a) 2D view of the final pose constraint network.

(b) Zoomed view of inset.

Figure 4-11 The triangulated structure for the RMS Titanic as computed from the final VAN pose estimate (Fig. 4-9(e)) and saved pairwise correspondences. Triangulated 3D points are defined as the midpoint of the minimum perpendicular distance between two corresponding camera rays. Since structure is triangulated on a pairwise basis, redundant 3D points may occur. (a) A histogram of the triangulation error (i.e., the minimum perpendicular distance) for all points across all established camera pairs. Note that the histogram contains two measures and that its y-axis has been clipped to show fine detail. The first measure (white) is the triangulation error based upon the relative-pose camera measurements used by the ESDF filter. This should serve as a baseline for the best possible pairwise triangulation error since each pose measure is the product of a two-view bundle adjustment. The second measure (black) is the triangulation error based upon the final VAN estimated poses. Scale for both measures has been set by the VAN estimate. Note that the VAN triangulated errors are more widely distributed than the pairwise bundle-adjusted poses. However, this is to be expected since VAN's global estimate takes into account all measured camera constraints. (b) The VAN triangulated points rendered in 3D (467,512 points in total). The "outliers" are due to poor triangulation resulting from residual error in the global VAN estimate. Again, this error is to be expected since VAN is not directly enforcing structure consistency, only pose consistency. In fact, because VAN is enforcing only pose consistency, the overall coherence of the point cloud corroborates the global consistency of VAN's pose estimates. (c)–(d) These plots display the same data as (a)–(b), but for a reduced set of triangulated data (363,799 points). This reduced set corresponds to throwing away all points having a triangulation error greater than 7.5 cm.



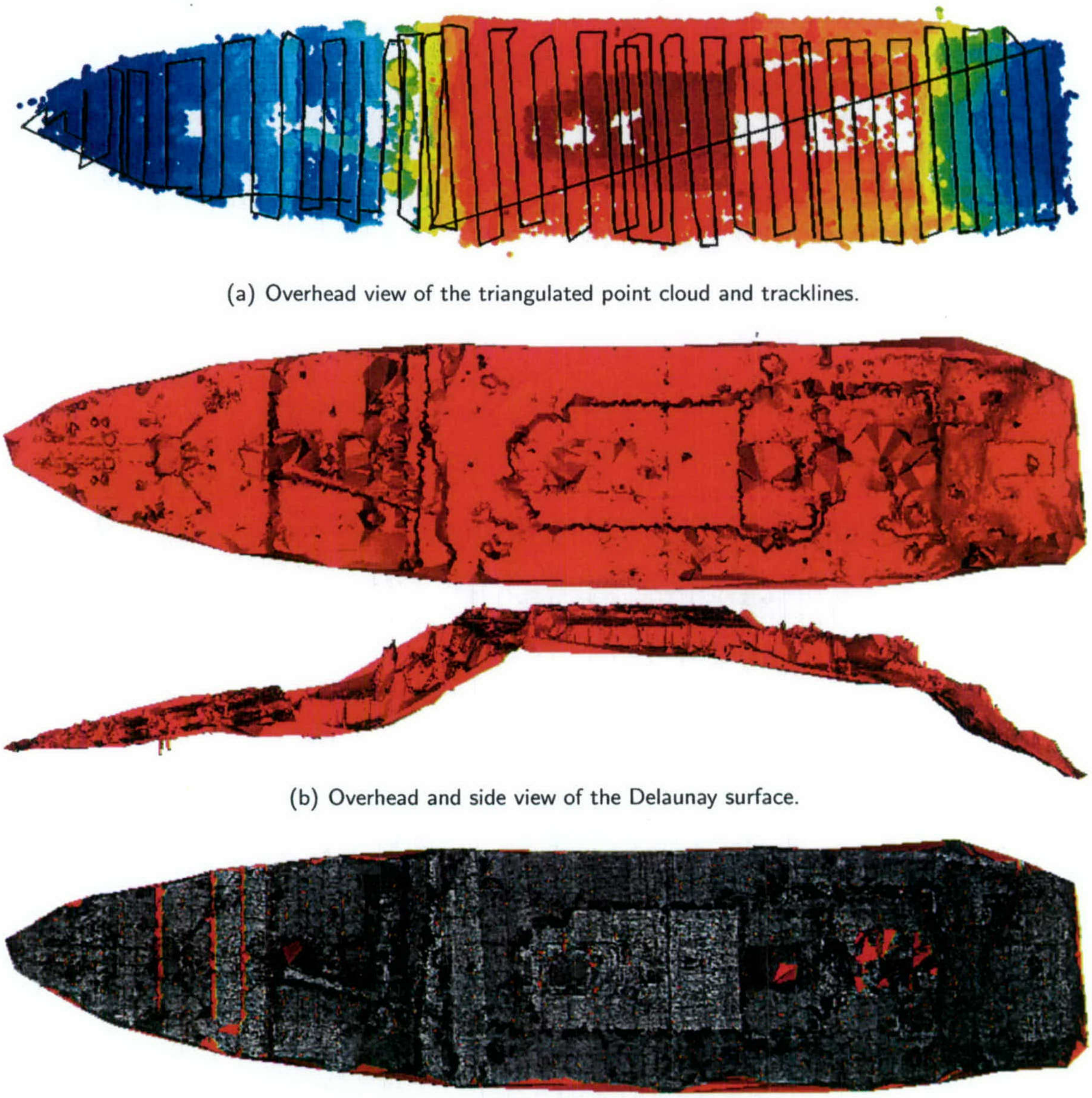
(a) Raw triangulation error.

(b) Raw point cloud.

(c) Reduced set triangulation error.

(d) Reduced set point cloud.

Figure 4-12 The triangulated point cloud, resulting Delaunay surface, and texture mapped rendering for the RMS Titanic. (a) The (reduced set) triangulated point cloud calculated using VAN pose estimates and pairwise correspondences (same color scale as in Fig. 4-11). Overlaid in black are the tracklines connecting sequential poses. (b) The resulting Delaunay triangulated surface. (c) The textured mapped surface as computed by back-projecting the images onto the Delaunay mesh (the tiling artifact is due to the simple overlay of images without blending). The red regions are places where no camera footprints back-project to the mesh. In particular, the red strips along the bow correspond to missing cross-track camera constraints (due to low overlap) in the final pose-network (Fig. 4-10). Note that these regions really should have texture, but that due to a lack of camera constraints there is residual error in this portion of the pose-network (however, the uncertainty ellipses of Fig. 4-10 reflect this).

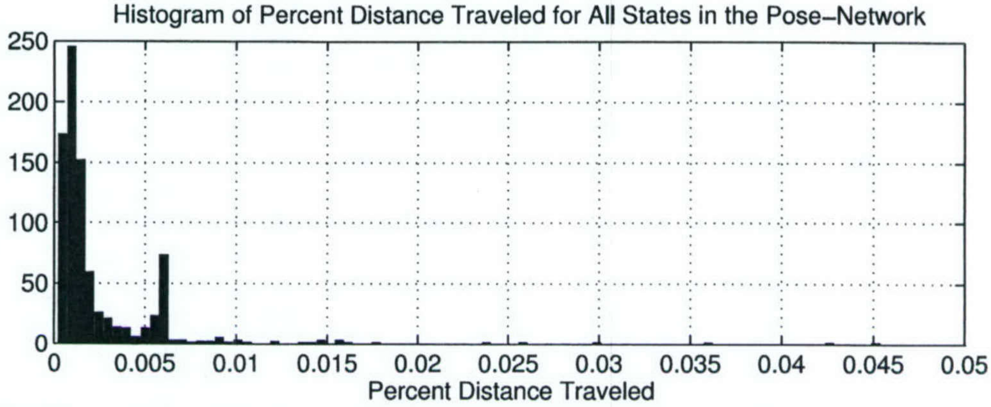


(a) Overhead view of the triangulated point cloud and tracklines.

(b) Overhead and side view of the Delaunay surface.

(c) Overhead view of the texture mapped surface.

Figure 4-13 A histogram of VAN's estimated precision, expressed as a percentage of distance traveled, for the RMS Titanic pose-network.



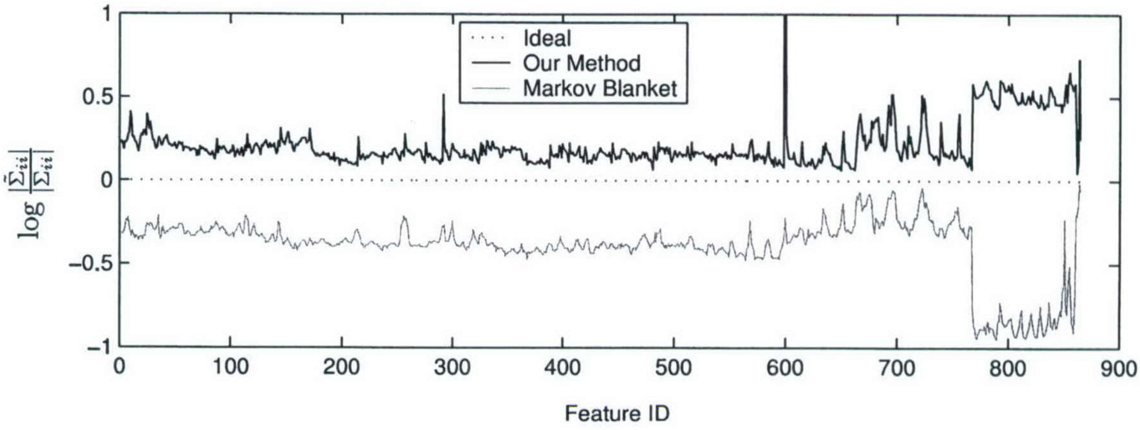
15.5 cm). While the actual face value of this number must be taken with a grain of salt (since there is no ground-truth with which to verify the filter's consistency), it does stress the point that closed-loop camera feedback provides a robust and improved navigation estimate even despite persistent dropouts in the bottom-lock Doppler measurement.

Experimental Results: Covariance Recovery

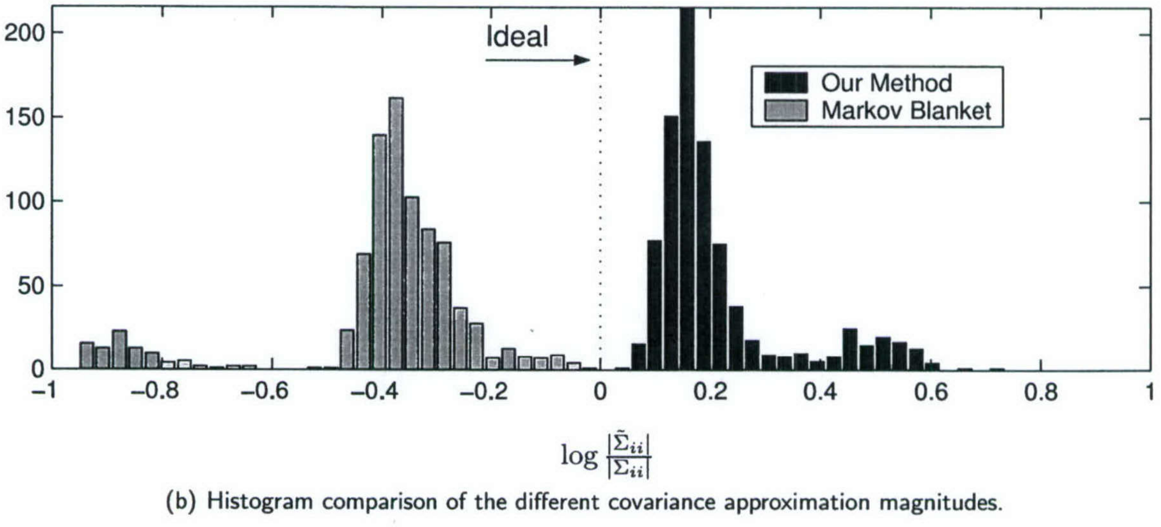
Fig. 4-14 provides a quantitative assessment comparing the covariance bounds obtained by our algorithm to the bounds obtained by inverting only the Markov blanket (§4.3.1). To provide a fair assessment, we choose to evaluate the *relative* uncertainty between the robot, $\mathbf{x}_r$, and any other map element, $\mathbf{x}_i$. Our justification for this metric is that the Markov blanket results in a conditional covariance that does not accurately reflect *global* map uncertainty, but rather *relative* map uncertainty. Using the information matrix of Fig. 4-1, for each map element, $\mathbf{x}_i$, we computed the first-order relative-pose covariance matrix between it and the robot. For our metric, we chose to compute the log of the determinant of the approximation covariance to the determinant of the actual covariances obtained by matrix inversion. Hence, ratios greater than one (conservative) are positive, and ratios less than one (overconfident) are negative. Note that Fig. 4-14 highlights that our method is conservative while the Markov blanket is overconfident. Furthermore, for this dataset the histogram shows that our method tends to be conservative by a smaller margin.

Finally, Fig. 4-15 and Fig. 4-16 both demonstrate the actual value of this conservative approximation within the context of pose-constrained correspondence searches. In particular, each figure shows a candidate pair of underwater images and the predicted epipolar geometry instantiated from our state estimate. Recall that for a calibrated camera, the epipolar geometry is specified by the relative camera pose and defines a 1D search constraint [64]. However, when the relative-pose estimate is uncertain, this 1D search constraint becomes a search *region* (§2.4.3). Fig. 4-15 depicts a case where the Markov blanket approximation fails due to its overconfident covariance estimate. This failure is indicated by the fact that its 6-sigma confidence search region does not contain the true correspon-

Figure 4-14 A quantitative comparison of the different covariance recovery techniques using the information matrix of Fig. 4-1. These plots compare the Markov blanket covariance approximation to the results of our method, both of which are shown relative to the actual covariance obtained by matrix inversion. For each method and state entry $\mathbf{x}_i$, we compute its relative-pose to the robot $\mathbf{x}_r$ (i.e., $\mathbf{x}_{ri} = \ominus\mathbf{x}_r \oplus \mathbf{x}_i$) and associated first-order covariance. We then plot the log of the ratio of the determinant of the approximated covariance to the determinant of the actual covariance to facilitate comparison of conservativeness (positive values) versus overconfidence (negative values). (a) Plot of the log ratio verses feature id for all $\mathbf{x}_i$. Note that a value of zero is ideal as this would indicate a ratio of one. (b) Same data as above but presented in histogram form. Both plots show that the method presented in this paper is conservative while the Markov blanket method is overconfident.



(a) Plot comparison of the different covariance approximation magnitudes.



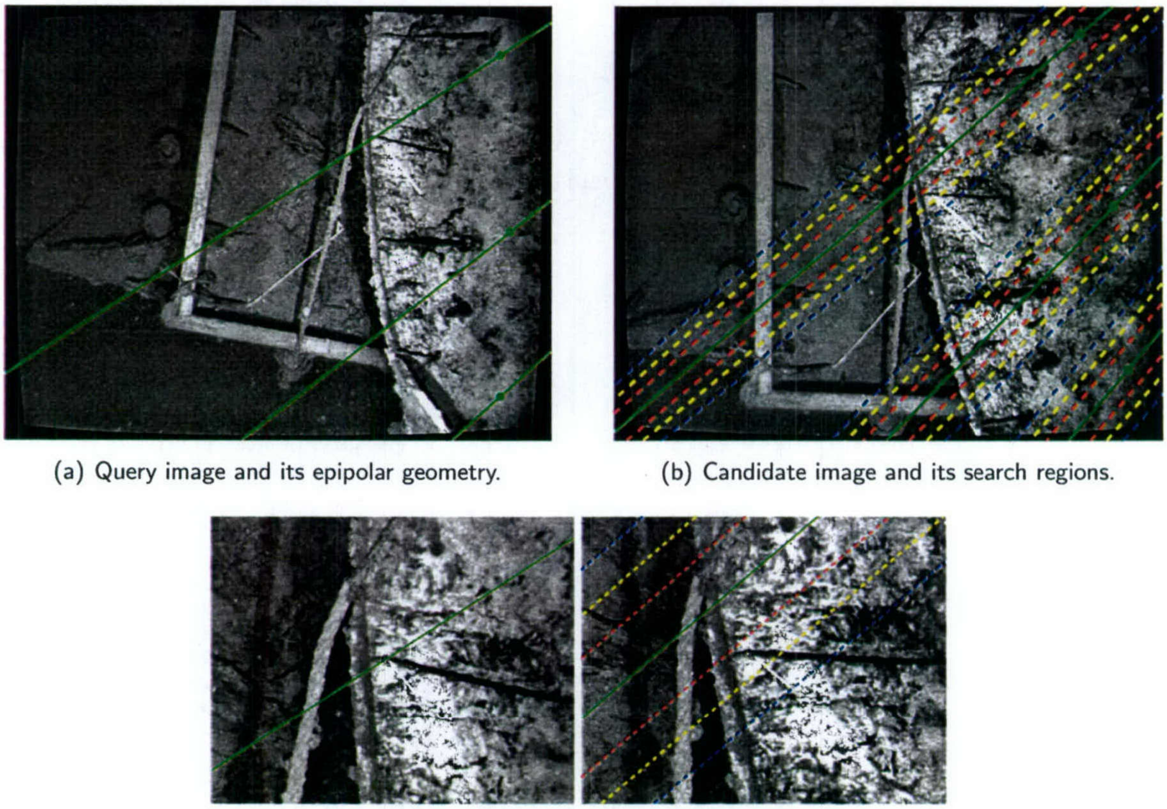
(b) Histogram comparison of the different covariance approximation magnitudes.

dence while, on the other hand, the regions computed by the actual covariance and our conservative approximation both do. Hence, for this image pair, putative correspondence establishment fails under the Markov blanket approximation, but meanwhile succeeds with ours. Additionally, Fig. 4-16 highlights that the amount of overconfidence in the Markov blanket approximation is unpredictable, since for a different image pair it produces comparable results.

4.6 Chapter Summary

In conclusion, this chapter showed that the delayed-state view-based SLAM information matrix is exactly sparse and, furthermore, that this sparsity is a direct consequence of the process model's Markovity. Moreover, while the covariance formulation requires quadratic storage, the number of nonzero off-diagonal elements in the ESDF information matrix is *linear* in the number of measured relative-pose constraints. This sparse matrix structure allows for $\mathcal{O}(n)$ full state recovery via recently proposed multilevel relaxation methods, while approximate partial state recovery allows motion prediction and navigation updates to be performed in constant time. Additionally, we also presented a novel algorithm for extracting consistent marginal covariance bounds from SLAM information filters. These bounds provide a conservative covariance approximation useful for real-world tasks such as image link hypothesis and pose-constrained correspondence searches. Furthermore, the technique's complexity scales asymptotically linear with map size. Finally, as a crowning achievement, we demonstrated results for the largest underwater visually navigated dataset to date using data collected from the RMS Titanic.

Figure 4-15 Performance of the different covariance recovery techniques within the context of image registration. In this example, the Markov blanket approximation fails, while both the actual covariance and our conservative approximation succeed. (a)–(b) These images are a proposed candidate pair for image registration. Image (a) represents the query image as viewed by the robot, and overlaid on top is the predicted epipolar geometry (green) instantiated from our state estimate. Image (b) is the proposed candidate for image registration, and overlaid on top are the pose-constrained correspondence search regions for 6-sigma confidence bounds. The different colored regions correspond to the three covariance recovery methods presented in this chapter: 1) our conservative method (blue), 2) the actual covariance based upon inverting the information matrix (yellow), and 3) the Markov blanket technique (red). (c) These images show a zoomed view of the true correspondence (indicated by the white arrows in (a)–(c)). Careful inspection reveals that the Markov blanket search region (red) does not contain the true correspondence. In contrast, both the actual covariance (yellow) and our covariance approximation (blue) do.

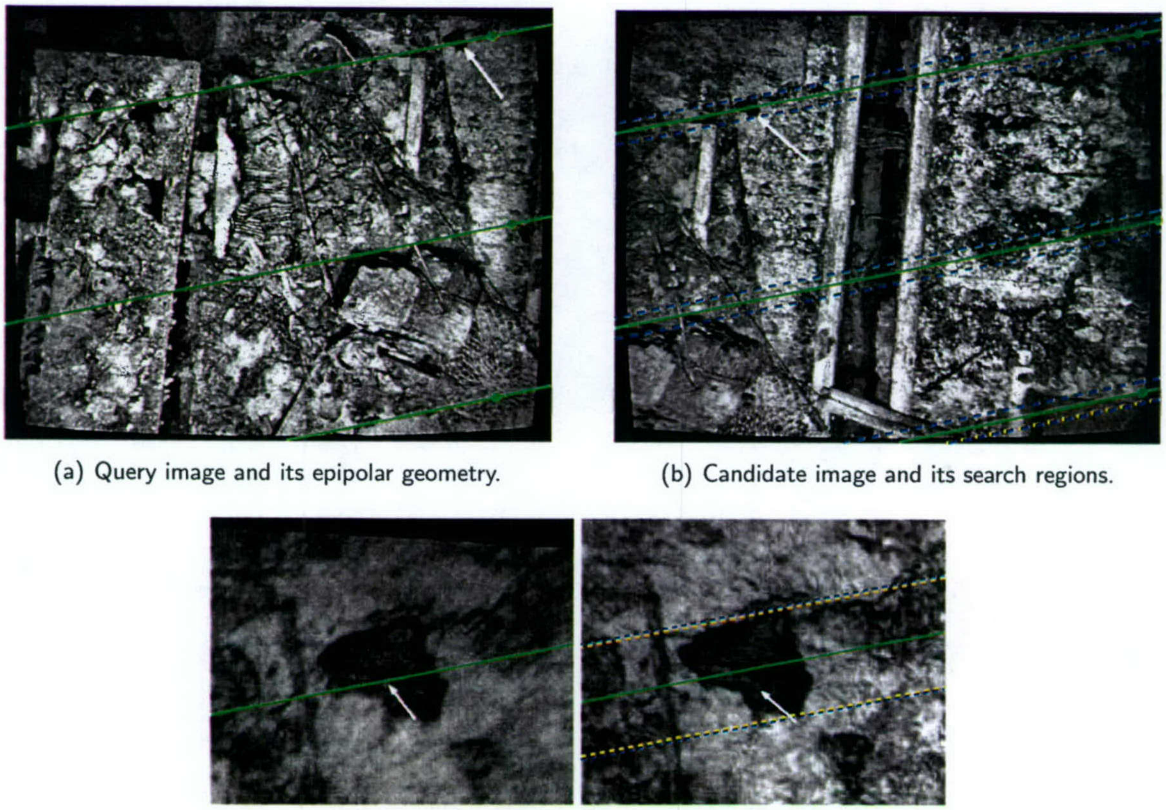


(a) Query image and its epipolar geometry.

(b) Candidate image and its search regions.

(c) Zoomed view. For this case, the Markov blanket approximation fails.

Figure 4-16 This figure depicts the same demonstration as Fig. 4-15, but for a different image pair. In this example, all three methods produce comparable results. This highlights the unpredictable nature of the Markov blanket approximation.



(a) Query image and its epipolar geometry.

(b) Candidate image and its search regions.

(c) Zoomed view. For this case, all three methods are comparable.

CHAPTER 5 Conclusions

In pursuing a VAN methodology, this thesis has advanced the current state-of-the-art in large-area underwater visual navigation. As evidence for this claim, we demonstrated successful automatic end-to-end processing for the largest visually navigated underwater dataset to date using data collected from the RMS Titanic (866 images, survey path length over 3 km, and 3100 m^2 of mapped area).

5.1 Contributions

This thesis has made general contributions to the understanding of scalable SLAM algorithms and systems-level computer vision. In particular, the contributions of this thesis are:

1. *We presented a systems-level, vision-based, 6-DOF SLAM framework (Chapter 2) that exploits the additional sensor capabilities of a calibrated pose-instrumented platform.*

   This top-down systems-level approach allowed us to overcome many of the challenging peculiarities associated with an underwater environment (e.g., unstructured natural terrain, low-overlap imagery, moving light source) by exploiting *a priori* platform information wherever possible (e.g., bounded error attitude measurements, scene depth constraints from altimeter). Additionally, we showed how pairwise relative-pose measurements can be recursively incorporated and fused with navigation data within a delayed-state EKF SLAM framework.

2. *We developed a novel pose-constrained correspondence search (Chapter 2) that incorporates* a priori *relative-pose information (e.g., sensor-based or posterior-based) to restrict the correspondence search to probable regions between hypothesized candidate pairs.*

   This search constraint is based upon a two-view point transfer relation that incorporates constraints on both relative-pose and scene depth (e.g., from an altimeter). This greatly improves the robustness of putative matching within a feature-based image registration methodology.

3. *We presented a theoretical investigation (Chapter 3) into the constant-time sparsification approximation employed by sparse extended information filters for feature-based SLAM.*

   In particular, we offered novel insight into the effect sparsification has on the consistency of the SLAM information posterior. Additionally, we also derived a modified sparsification rule that maintains sparsity while yielding results comparable to the standard full-covariance EKF approach. However, this accuracy comes at the loss of constant-time computation.

4. *We presented the novel insight that a view-based SLAM framework retains exact sparsity when posed in the information form (Chapter 4) and, therefore, can exploit the sparse posterior representation in a consistent way.*

   This result lead to the development of exactly sparse delayed-state filters as a consistent, scalable, recursive, fusion framework for incorporating relative-pose measurements. Furthermore, ESDFs require only linear storage and at most $\mathcal{O}(n)$ complexity per update where n is the number of map images. Additionally, if relaxation is performed over multiple time-steps, this complexity can be traded off for slightly increased state recovery error.

5. *Lastly, we developed a novel algorithm (Chapter 4) for efficiently accessing and maintaining consistent covariance bounds within a SLAM information filter thereby greatly increasing the reliability of data association.*

   The foundation of this algorithm is the solution of a sparse linear system coupled with the application of constant-time Kalman updates. Its output is a consistent set of marginal covariances estimates suitable for robot planning and data association.

5.2 Failure Modes

While within the context of this thesis we have demonstrated that our large-area VAN framework has significant advantages over traditional underwater localization methods, we are compelled to also point out what we think are the weak points of this framework. In particular, we believe failure modes of VAN to be:

1. *False positive image registrations and/or repetitive scene structure.*

   VAN's closed-loop navigation feedback is derived from the registration of image pairs with common scene overlap. Since these pairwise camera measurements are fed to the delayed-state filter as a measurement of relative-pose modulo scale, a false positive camera measurement could have devastating consequences for the updated pose-network topology. In particular, a false image registration could cause the state estimate to converge to an incorrect trajectory. Though utilizing a robust image registration methodology reduces the likelihood of a false positive within the VAN framework, it does not protect against an environment with repetitive scene structure.

2. *Linearization error and filter divergence.*

   Recall that VAN represents the posterior distribution using its first two moments under a first-order linearized approximation evaluated about the current state mean. The danger with this approach is that if the current estimate "drifts too far from the truth", then the linearization point in state space may no longer be valid and could ultimately lead to filter divergence. One method for avoiding this condition is to keep linearization error small by maintaining good map contact so that, in practice, the state estimate never drifts too far from the truth (for example, typical grid-based AUVs surveys achieve this).

5.3 Future Work

This thesis has laid the foundation for a promising infrastructure-free, near-seafloor, navigation strategy complementing the low-overlap exploratory surveys typical of AUVs. However, as always, there is room for additional improvement. Further areas of research that we have identified as warranting future investigation are:

1. *Develop a fast lookup-table method for initial screening of hypothesized image pairs.*

   Since we are using a view-based framework, raw images must be registered to extract camera motion. For this purpose, potential overlapping candidate pairs are first proposed based upon probable spatial proximity, and then verified by attempting robust image registration (e.g., RANSAC, LMedS). The limitation of this strategy is that pairwise image registration is arguably the slowest component of the VAN framework. This suggests that we can potentially gain an increase in efficiency by being more selective in the candidate image pair proposal stage. Ideally, characteristics the pre-screening method should posses are:

   - have a near-zero probability of missed detection so that overlapping candidate image pairs are not passed over for attempted registration,
   - have a low false alarm rate so that non-overlapping candidate image pairs are not recommended for attempted registration,
   - be computationally cheap.

   As a possible suggestion, the method could employ a look-up table strategy computed on feature point projective invariants (e.g., cross-ratios [64]).

2. *Develop a metric for computing the quality of feature content within an image.*

   As we saw with the Stellwagen Bank dataset of Chapter 2, seafloor topography can vary widely even within a single survey area (e.g., from a featureless muddy bottom to large boulders). Hence, the question is "should all images be considered equal for inclusion in our delayed-state vector?" For re-localization purposes we argue no. While we can use our navigation sensors to give us a good prior for registering sequential images of say a featureless sandy bottom, this type of imagery does little for us in terms of re-localizing the robot after a large survey loop. Therefore, it would be beneficial to develop a metric that distinguishes images based upon their feature content so that we can avoid including "featureless" images into our view-based map.

3. *Extend the concept behind pairwise pose-constrained correspondence searches to work with multiple image candidate sets.*

   Loop-closing image registration is one of the most difficult challenges of the VAN framework (any SLAM framework really). When a good prior exists for the camera motion, we showed in §2.4.3 how to exploit a pairwise pose-constrained correspondence search to greatly improve the robustness of putative correspondence selection. Unfortunately, when closing a very large loop, the accumulated uncertainty implies that our global pose prior will essentially be useless on a pairwise basis for restricting the putative search space. However, if we consider extending the putative matching to *collections* of images, then we can instead perform a joint putative matching across sets that exploits the fact that our relative-pose prior is well known intra-set. The main idea would be to look for matches that are mutually consistent across the two sets (similar to the joint compatibility data association strategy of Neria and Tardos [118]). This should increase the "signal-to-noise ratio" for putative matching since feature similarity measures could be considered in aggregate as illustrated by Fig. 5-1.

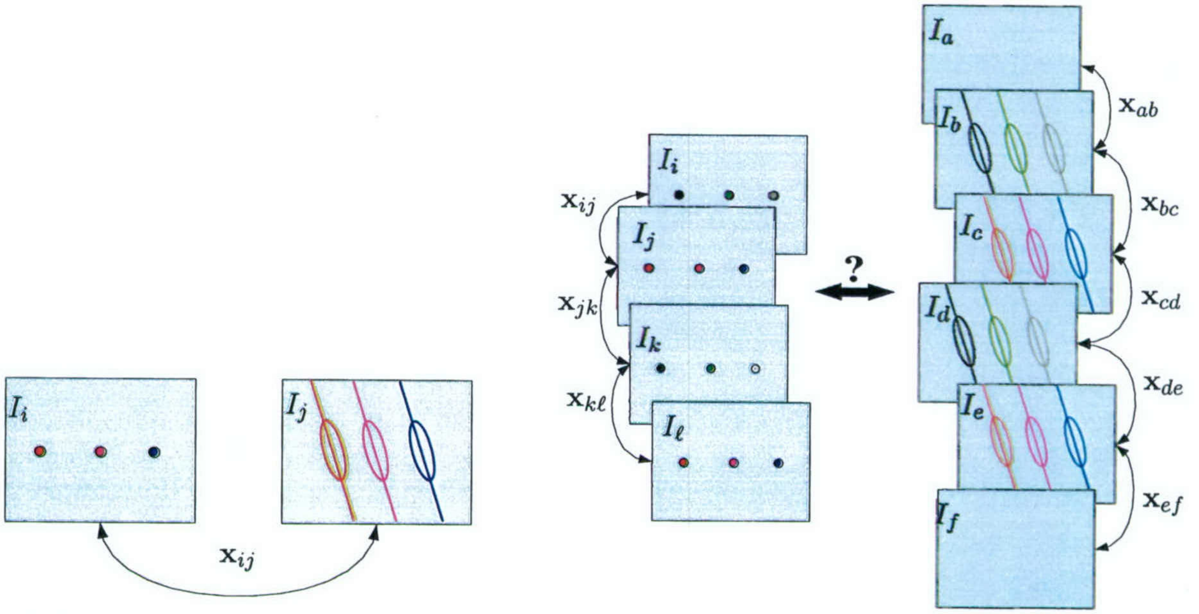
4. *Extend the information filter covariance recovery algorithm to be less conservative.*

   Our covariance recovery algorithm of §4.3.2 for SLAM information filters guarantees that the marginal covariance estimates are consistent with respect to the actual covariances obtained by matrix inversion. However, because we only update our covariance bounds on a per re-observation basis, they can become very conservative if we go for long periods without re-observing. For example, the worst cast scenario occurs when closing a very large loop since only the bounds within the viewable vicinity are updated. Hence, this suggests that it would be beneficial to extend our update strategy to include more than just the current state under view by propagating the observation knowledge throughout the constraint network. Unfortunately, not having access to the cross-covariances complicates things. As a possible suggestion, maybe a hybrid strategy using covariance intersection and the fixed-size Kalman updates of §4.3.2 would be fruitful?

5. *Lastly, develop a way for not having to add additional robot poses to the ESDF state vector once we've collected enough views to sufficiently characterize an area.*

   One criticism of the ESDF view-based framework is that exact sparsity in the information matrix can only be achieved by perpetually adding robot poses over time. For example, consider the case where we've collected enough "views" to sufficiently characterize an area. In this scenario, when we return to a previously mapped area, rather than adding more states, instead we should be able to just localize with respect to the finite collection we already have. However, the problem with this strategy is that just like in feature-based SLAM, if we begin to marginalize out our robot trajectory the information matrix will densify as illustrated by Fig. 5-2. Therefore, if we want to restrict our representation to environment size (fixed) and not time (unbounded),

Figure 5-1 A multi-image joint-correspondence search. (a) The pairwise pose-constrained correspondence search of §2.4.3 exploits the relative-pose prior $\mathbf{x}_{ij}$ between images I_i and I_j to reduce the putative correspondence search space. (b) For loop-closing, the accumulated uncertainty along the loop may render our global pose prior useless. However, we should still be able to exploit the fact that our relative-pose information is accurate for small *sequences* of images. For example, given the overlapping collection of images on the left, the relative-pose information $\mathbf{x}_{ij}$, $\mathbf{x}_{jk}$, and $\mathbf{x}_{k\ell}$ should allow us to extend our correspondence search across multiple raw frames in our view-based map as shown on the right. In other words, instead of just relying upon the pairwise discriminatory power of our feature descriptor, we can use our relative-pose information to look for mutually consistent putative matches across collections of images by exploiting the knowledge that if I_ℓ and I_e overlap, then so should I_k and I_d. Effectively, this should increase the "signal-to-noise ratio" of our feature similarity measure as putative matches would be considered jointly across images and checked for mutual consistency. This should in turn make visually-based loop-closing both more tractable and more robust.

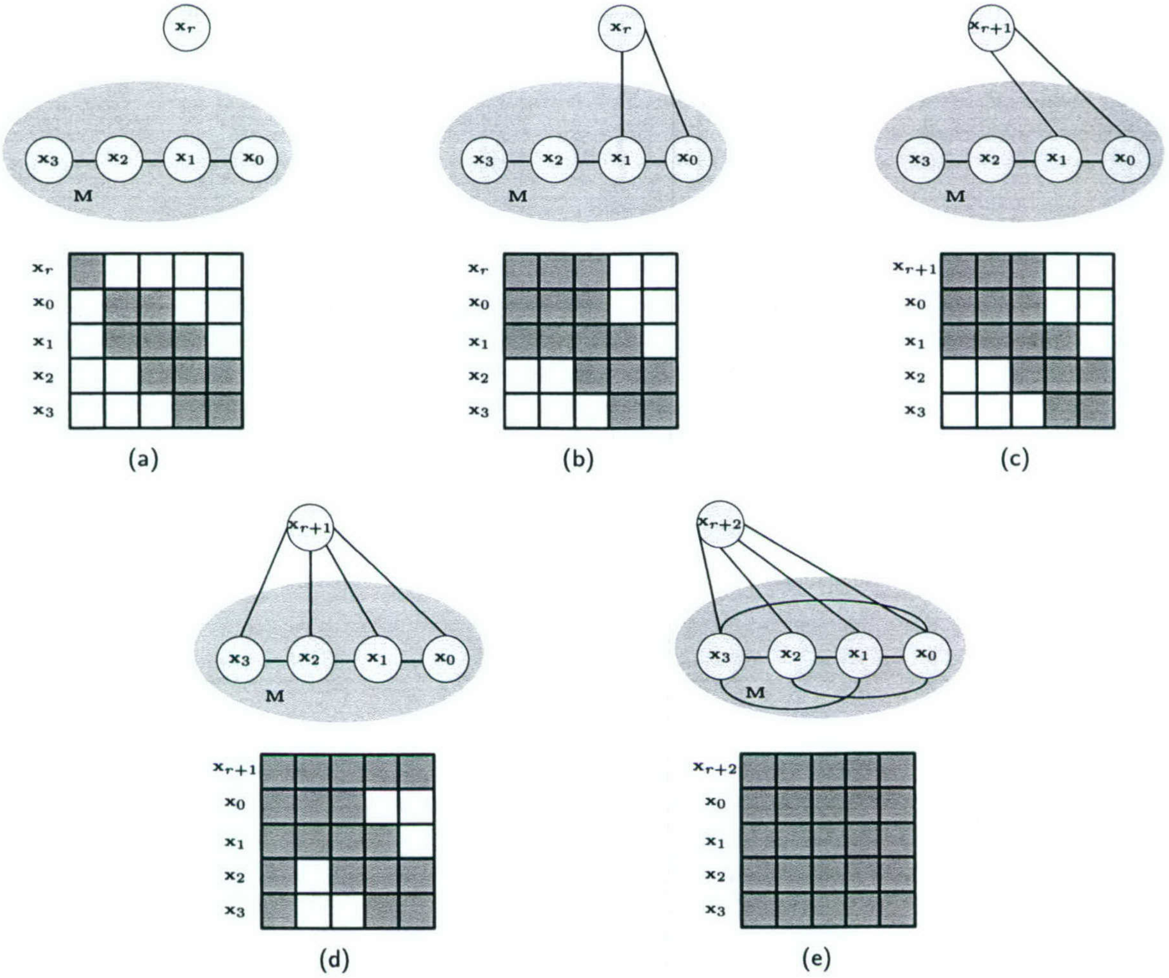


(a) Standard pairwise pose-constrained correspondence search.

(b) Multi-image joint-putative constrained correspondence search.

then this suggests that some sort of approximation is required (similar to the pruning strategies employed by feature-based SLAM information filters). Hence, further research into a consistent, general, computationally efficient, edge pruning algorithm may be fruitful for all areas of SLAM.

Figure 5-2 Localizing in a previously mapped area fills in the ESDF information matrix unless we continue to add robot poses to our representation. (a) Suppose our view-based map consists of the set of poses $\mathbf{M} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and the robot, $\mathbf{x}_r$, returns to this previously mapped area. For both simplicity and clarity, assume that the robot doesn't have any shared information with the map as shown. (b) Now suppose that the robot re-localizes itself by making relative-pose measurements $\mathbf{x}_{r0}$ and $\mathbf{x}_{r1}$. (c) Suppose that rather than augmenting our map representation to include the robot pose, $\mathbf{x}_r$, we instead perform time-prediction (i.e., the robot state evolves from $\mathbf{x}_r$ to $\mathbf{x}_{r+1}$). (d) Now suppose that the robot makes relative-pose measurements $\mathbf{x}_{r2}$ and $\mathbf{x}_{r3}$. (e) Time propagating the robot pose $\mathbf{x}_{r+1}$ to $\mathbf{x}_{r+2}$, followed by marginalization over $\mathbf{x}_{r+1}$, results in a fully dense information matrix.



APPENDIX A

Robot System, Models, and Coordinate Frames

THE aim of this thesis is to develop a novel, scalable, SLAM algorithm that respects the constraints of low-overlap imagery typical of AUVs while exploiting the information associated with the inertial sensors that are routinely available on such platforms. Developing such a framework requires that we make judicious assumptions regarding the type of platform capabilities that can be reasonably expected, the set of conventions for defining coordinate frame relationships, the level of detail required in modeling vehicle dynamics, and the type of information measured by our strap-down sensors. In this chapter we describe our platform assumptions, coordinate frame conventions, and vehicle/sensor models.

A.1 Platform

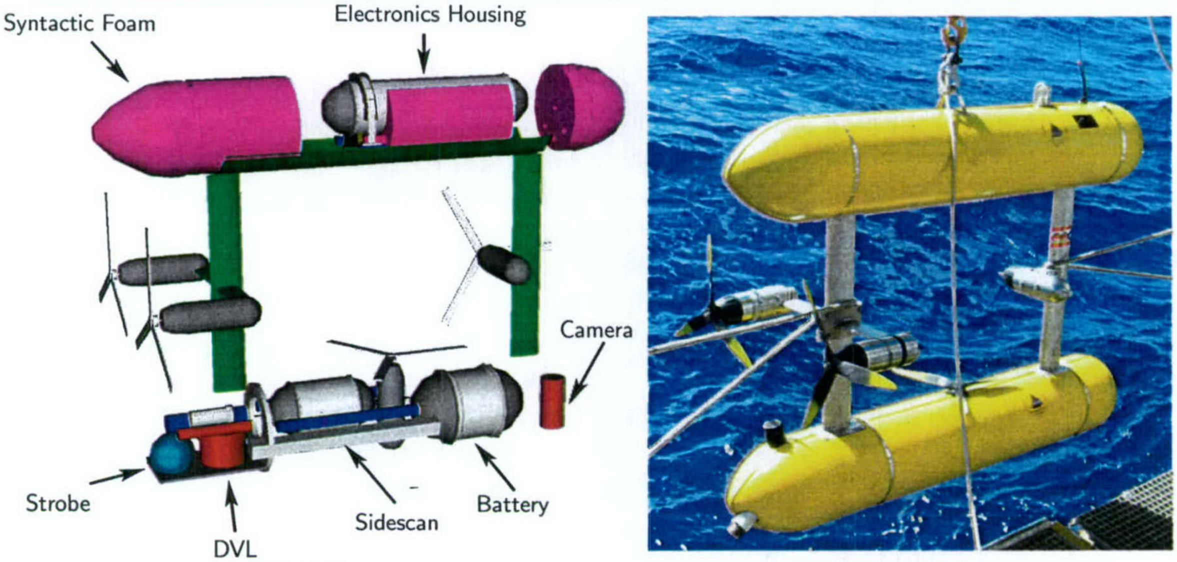
The assumed platform capabilities are thoroughly grounded in modern oceanographic AUV technology. Much of the experimental development work for this thesis has been conducted using the SeaBED AUV [145–147] — a bottom-following, hover-capable, imaging research platform (Fig. A-1) equipped with a standard suite of underwater dead-reckoned navigation sensors (Table A.1). In particular, its main navigation source is an acoustic Doppler velocity log which measures seafloor referenced velocities with a precision on the order of 1–2 mm/s. These velocities can then be integrated over time to provide a dead-reckoned position estimate for real-time control. The accuracy of the position estimate is governed by absolute orientation measurements. In our case, these are minimally instrumented using a magnetic-compass for heading (good to a few degrees), and tilt sensors for roll and pitch (±0.5°). In addition to the DVL estimate, bounded error vehicle depth is measured via a Paroscientific pressure sensor good to 0.01% precision, which translates to a few centimeters over full ocean depths.

Since SeaBED is intended to be a scientific imaging platform, its uses a two-hull design to give good separation between center of mass and center of buoyancy making the vehicle passively pitch and roll stable. For optical imaging, SeaBED is equipped with a calibrated,

Table A.1 Specifications of the SeaBED AUV platform.

Vehicle	Depth rating	2000 m	
	Size	2.0 m (L) × 1.5 m (H) × 1.5 m (W) (bbox)	
	Mass	200 kg	
	Cruising Speed	0.15–1.2 m/s	
	Batteries	2 kWh Li-ion pack	
	Propulsion	(4) 150 W brush-less DC thrusters	
Navigation	Depth	0.01%	Paroscientific pressure sensor
	Velocity	±1–2 mm/s	RDI 1200 kHz Navigator ADCP
	Tilt	±0.5°	RDI (internal)
	Heading	±2.0°	RDI (internal)
	Altitude	0.1 m	RDI (beam avg.)
	Angular rates	1°/s	Crossbow AHRS
Optical Imaging	Camera	12bit 1280×1024 Pixelfy CCD (bw or color)	
	Lighting	(1) 200 W · s strobe	
	Separation	1 m between camera and light	
Acoustic Imaging	Sidescan sonar	300 kHz MST (300 m depth rating)	
	Pencil-beam sonar	675 kHz Imagenex 881	
Other Sensors	CTD	Seabird 37SBI	

Figure A-1 The SeaBED AUV



(a) CAD model.

(b) Vehicle with skins.

down-looking, high dynamic range (12bit) color CCD that can be swapped out for a black and white camera depending on the scientific application. SeaBED relies upon a 2 kWh Li-ion battery pack for power and, therefore, uses strobe flash photography to reduce power consumption and increase vehicle endurance. Typical survey speeds are usually in the range of 20–60 cm/s, though, the vehicle is capable of obtaining speeds up to 1.2 m/s. The former range of speeds provide approximately 15–35% temporal image overlap at altitudes ranging from 2.5–4.0 m.

A.2 6-DOF Coordinate Frame Relationships

This section describes the relevant reference frames involved in vehicle navigation and their 6-DOF coordinate frame relationships as illustrated in Fig. A-2. We follow standard SNAME[1] convention [48] and define the vehicle frame, denoted subscript v, to be coincident with a fixed point on the vehicle and oriented such that the positive $\textsc{x}_v$-axis is aligned with the bow, positive $\textsc{y}_v$ starboard, and $\textsc{z}_v$ down completing a right handed coordinate frame.

Additionally, we must consider each onboard sensor's own internal coordinate frame in which measurements are expressed and its relationship to the vehicle. The sensor frame, denoted subscript s, is assumed to be static and known with respect to the vehicle frame (i.e., calibrated beforehand). Finally, two navigation frames are defined and used for expressing vehicle pose. The first is the world frame, denoted subscript w, which is a static reference frame located at the water surface oriented with $\textsc{x}_w$-East, $\textsc{y}_w$-North, and $\textsc{z}_w$-Up and is useful for displaying results since it follows standard map convention. The second navigation frame that we define is the local-level frame, denoted subscript ℓ. This frame is coincident with the world frame, however, it is oriented with $\textsc{x}_\ell$-North, $\textsc{y}_\ell$-East, $\textsc{z}_\ell$-Down and corresponds to a zero-orientation (i.e., local-level) version of the vehicle frame; this frame is useful for navigation because standard compass-measurements are consistent with the right-hand rule convention about the $\textsc{z}$-axis.

A.2.1 Pose Description

We adopt the Smith, Self, and Cheeseman [154] coordinate frame notation and define the 6-DOF pose of frame j with respect to frame i as

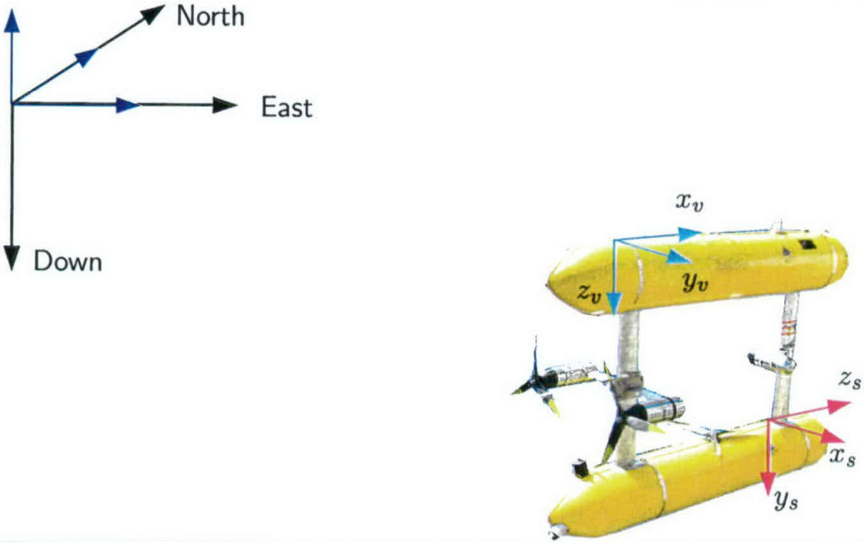
$$\mathbf{x}_{ij} = \left[{}^{i}\mathbf{t}_{ij}{}^{\top}, \mathbf{\Theta}_{ij}^{\top}\right]^{\top} = [x_{ij}, y_{ij}, z_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}]^{\top}.$$

Here, ${}^{i}\mathbf{t}_{ij}$ is a Euclidean 3-vector from i to j as expressed in frame i, and $\mathbf{\Theta}_{ij}$ is a 3-vector of XYZ-convention roll, pitch, heading Euler angles.[2] For this Euler definition the 3×3

[1] The Society of Naval Architects and Marine Engineers.

[2] Note that we differ from Smith, Self, and Cheeseman in our roll, pitch, heading (RPH) Euler angle definition and instead follow Fossen's [48], which is standard convention for guidance and control applications. Effectively, our RPH are swapped for their HPR, thus, the 6-DOF relationships presented in this section correspond to permutations of the relations given in the appendix of [154].

Figure A-2 Illustration of the relevant navigation frames. The blue and black frames represent the world and local-level frames respectively which are coincident. The cyan frame represents the vehicle reference frame while the magenta frame represents an arbitrary sensor frame. The sensor and vehicle frames are attached to the same rigid body and therefore are static with respect to each other.



orthonormal rotation matrix that rotates frame j into frame i is defined as

$$\begin{aligned}
{}^{i}_{j}\mathrm{R} &= \mathrm{rotxyz}(\boldsymbol{\Theta}_{ij}) \\
&= \mathrm{rotz}(\psi_{ij})^\top \, \mathrm{roty}(\theta_{ij})^\top \, \mathrm{rotx}(\phi_{ij})^\top \quad \text{(principle rotation sequence)} \\
&= \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}^\top \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}^\top \\
&= \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}.
\end{aligned}$$

Similarly, given the rotation matrix ${}^{i}_{j}\mathrm{R}$ we can recover the Euler angles as[3]

$$\begin{aligned}
\phi_{ij} &= \mathrm{atan2}\big({}^{i}_{j}\mathrm{R}_{1,3}\sin\psi_{ij} - {}^{i}_{j}\mathrm{R}_{2,3}\cos\psi_{ij}, \quad -{}^{i}_{j}\mathrm{R}_{1,2}\sin\psi_{ij} + {}^{i}_{j}\mathrm{R}_{2,2}\cos\psi_{ij}\big) \\
\theta_{ij} &= \mathrm{atan2}\big(-{}^{i}_{j}\mathrm{R}_{3,1}, \quad {}^{i}_{j}\mathrm{R}_{1,1}\cos\psi_{ij} + {}^{i}_{j}\mathrm{R}_{2,1}\sin\psi_{ij}\big) \\
\psi_{ij} &= \mathrm{atan2}\big({}^{i}_{j}\mathrm{R}_{2,1}, \quad {}^{i}_{j}\mathrm{R}_{1,1}\big)
\end{aligned}$$

where the notation ${}^{i}_{j}\mathrm{R}_{m,n}$ means element (m, n) of ${}^{i}_{j}\mathrm{R}$. Note that from the 6-vector pose description $\mathbf{x}_{ij}$, we can conveniently obtain the standard 4 × 4 homogeneous coordinate

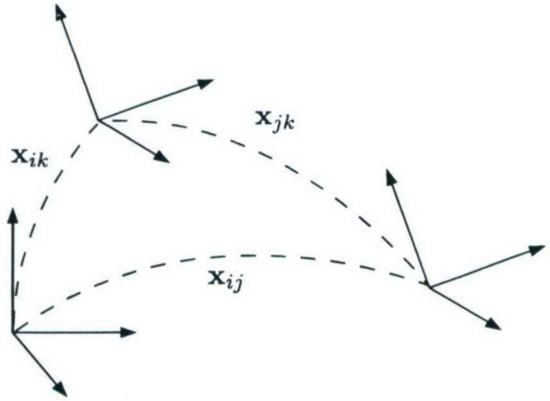
[3] Note that this representation of orientation suffers from a singularity at $\theta = \pm\frac{\pi}{2}$, but that in practice our vehicle application never operates anywhere near this singular configuration (i.e., pitch ±90°).

transformation matrix from frame j to frame i as

$$ {}^{i}_{j}\mathrm{H} = \begin{bmatrix} {}^{i}_{j}\mathrm{R} & {}^{i}\mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix}. $$

We can use this 6-vector representation to define the mechanics of some fundamental coordinate frame operations used as building-blocks in articulating more complex coordinate frame relationships. These operations are particularly useful in §A.4 where we define our strap-down sensor observation models.

Figure A-3 General coordinate frame relations for three arbitrary frames i, j, and k.



A.2.2 Head-to-Tail Operation

The head-to-tail relationship is a fundamental operation and is used to describe coordinate frame *composition.* Given pose vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$, the head-to-tail operation yields frame k with respect to frame i (i.e., $\mathbf{x}_{ik}$) as illustrated in Fig. A-3. Using standard homogeneous coordinate transforms we can derive this composition as

$$ {}^{i}_{k}\mathrm{H} = {}^{i}_{j}\mathrm{H}\,{}^{j}_{k}\mathrm{H} $$

from which the resulting transform ${}^{i}_{k}\mathrm{H}$ can be decomposed into ${}^{i}_{k}\mathrm{R}$ and ${}^{i}\mathbf{t}_{ik}$ to obtain the pose vector $\mathbf{x}_{ik}$. Similarly, we can sidestep this intermediate decomposition step and define this head-to-tail coordinate frame composition directly in component form as

$$ \begin{aligned} \mathbf{x}_{ik} &= \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} \\ &= [x_{ik}, y_{ik}, z_{ik}, \phi_{ik}, \theta_{ik}, \psi_{ik}]^{\top} \\ &= \begin{bmatrix} {}^{i}_{j}\mathrm{R} \begin{bmatrix} x_{jk} \\ y_{jk} \\ z_{jk} \end{bmatrix} + \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix} \\ \operatorname{atan2}\big({}^{i}_{k}\mathrm{R}_{1,3} \sin\psi_{ik} - {}^{i}_{k}\mathrm{R}_{2,3} \cos\psi_{ik},\;\; -{}^{i}_{k}\mathrm{R}_{1,2} \sin\psi_{ik} + {}^{i}_{k}\mathrm{R}_{2,2} \cos\psi_{ik}\big) \\ \operatorname{atan2}\big(-{}^{i}_{k}\mathrm{R}_{3,1},\;\; {}^{i}_{k}\mathrm{R}_{1,1} \cos\psi_{ik} + {}^{i}_{k}\mathrm{R}_{2,1} \sin\psi_{ik}\big) \\ \operatorname{atan2}\big({}^{i}_{k}\mathrm{R}_{2,1},\;\; {}^{i}_{k}\mathrm{R}_{1,1}\big) \end{bmatrix} \end{aligned} $$

where ${}^{i}_{k}\mathrm{R} = {}^{i}_{j}\mathrm{R}\,{}^{j}_{k}\mathrm{R}$.

The Jacobian is a useful quantity that allows us to compute a first-order covariance estimate of $\mathbf{x}_{ik}$ in the event $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$ are random variables.[4] For the head-to-tail relationship this is given by

$$\begin{aligned}
\mathrm{J}_{\oplus} &= \frac{\partial \mathbf{x}_{ik}}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jk})} \\
&= \begin{bmatrix} \mathrm{J}_{\oplus 1} & \mathrm{J}_{\oplus 2} \end{bmatrix} \\
&= \left[\begin{array}{cc|cc} \mathrm{I}_{3\times 3} & \mathrm{M} & {}^{i}_{j}\mathrm{R} & 0_{3\times 3} \\ 0_{3\times 3} & \mathrm{K}_1 & 0_{3\times 3} & \mathrm{K}_2 \end{array}\right]
\end{aligned}$$

where $\mathrm{J}_{\oplus 1}$ and $\mathrm{J}_{\oplus 2}$ correspond to the left and right half partitioning of $\mathrm{J}_{\oplus}$ (i.e., partials with respect to $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$ respectively) and

$$\begin{aligned}
\mathrm{M} &= \begin{bmatrix} {}^{i}_{j}\mathrm{R}_{1,3}\, y_{jk} - {}^{i}_{j}\mathrm{R}_{1,2}\, z_{jk} & (z_{ik} - z_{ij}) \cos\psi_{ij} & -(y_{ik} - y_{ij}) \\ {}^{i}_{j}\mathrm{R}_{2,3}\, y_{jk} - {}^{i}_{j}\mathrm{R}_{2,2}\, z_{jk} & (z_{ik} - z_{ij}) \sin\psi_{ij} & (x_{ik} - x_{ij}) \\ {}^{i}_{j}\mathrm{R}_{3,3}\, y_{jk} - {}^{i}_{j}\mathrm{R}_{3,2}\, z_{jk} & -x_{jk}\cos\theta_{ij} - \big(y_{jk}\sin\phi_{ij} + z_{jk}\cos\phi_{ij}\big)\sin\theta_{ij} & 0 \end{bmatrix} \\
\mathrm{K}_1 &= \begin{bmatrix} \cos\theta_{ij}\cos(\psi_{ik} - \psi_{ij})\sec\theta_{ik} & \sin(\psi_{ik} - \psi_{ij})\sec\theta_{ik} & 0 \\ -\cos\theta_{ij}\sin(\psi_{ik} - \psi_{ij}) & \cos(\psi_{ik} - \psi_{ij}) & 0 \\ {}^{j}_{k}\mathrm{R}_{1,2}\sin\phi_{ik} + {}^{j}_{k}\mathrm{R}_{1,3}\cos\phi_{ik}\sec\theta_{ik} & \sin(\psi_{ik} - \psi_{ij})\tan\theta_{ik} & 1 \end{bmatrix} \\
\mathrm{K}_2 &= \begin{bmatrix} 1 & \sin(\phi_{ik} - \phi_{jk})\tan\theta_{ik} & \big({}^{i}_{j}\mathrm{R}_{1,3}\cos\psi_{ik} + {}^{i}_{j}\mathrm{R}_{2,3}\sin\psi_{ik}\big)\sec\theta_{ik} \\ 0 & \cos(\phi_{ik} - \phi_{jk}) & -\cos\theta_{jk}\sin(\phi_{ik} - \phi_{jk}) \\ 0 & \sin(\phi_{ik} - \phi_{jk})\sec\theta_{ik} & \cos\theta_{jk}\cos(\phi_{ik} - \phi_{jk})\sec\theta_{ik} \end{bmatrix}.
\end{aligned}$$

Example:
Given vehicle pose in the local-level frame, $\mathbf{x}_{\ell v}$, we can compute the corresponding sensor pose in the local-level frame as

$$\mathbf{x}_{\ell s} = \mathbf{x}_{\ell v} \oplus \mathbf{x}_{vs}$$

where $\mathbf{x}_{vs}$ is the static sensor-to-vehicle pose. Assuming that vehicle pose $\mathbf{x}_{\ell v}$ is a random variable with covariance $\Sigma_{\mathbf{x}_{\ell v}}$, then to first-order the covariance of the sensor pose in the local-level frame is

$$\Sigma_{\mathbf{x}_{\ell s}} = \mathrm{J}_{\oplus 1} \Sigma_{\mathbf{x}_{\ell v}} \mathrm{J}_{\oplus 1}^{\top}$$

since $\mathbf{x}_{vs}$ is static and assumed *known* (i.e., $\mathbf{x}_{vs}$ is not a random variable).

Example A.1 Obtaining sensor pose and its uncertainty in the local-level frame via compounding.

A.2.3 Inverse Operation

The inverse relationship is another fundamental operation and is used for *reversing* a coordinate frame relationship. Given pose vector $\mathbf{x}_{ij}$, the inverse operation yields frame i with

[4] If $\mathbf{x}$ is a random variable with mean $\boldsymbol{\mu}_x$ and covariance Σ_{xx}, then to first-order the random variable $\mathbf{y} = \mathbf{f}(\mathbf{x})$ has mean $\boldsymbol{\mu}_y = \mathbf{f}(\boldsymbol{\mu}_x)$ and covariance $\Sigma_{yy} = \mathrm{J}\Sigma_{xx}\mathrm{J}^{\top}$ where $\mathrm{J} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\big|_{\mathbf{x}=\boldsymbol{\mu}_x}$ [154].

respect to frame j, (i.e., $\mathbf{x}_{ji}$). Using standard homogeneous coordinate transform notation we can derive this operation as

$${}^{j}_{i}\mathrm{H} = {}^{i}_{j}\mathrm{H}^{-1}$$

from which the resulting transform ${}^{j}_{i}\mathrm{H}$ can be decomposed into ${}^{j}_{i}\mathrm{R}$ and ${}^{j}\mathbf{t}_{ji}$ to obtain the pose vector $\mathbf{x}_{ji}$. Again, we can define this coordinate frame operation directly in component form as

$$\begin{aligned}
\mathbf{x}_{ji} &= \ominus\mathbf{x}_{ij} \\
&= [x_{ji}, y_{ji}, z_{ji}, \phi_{ji}, \theta_{ji}, \psi_{ji}]^\top \\
&= \begin{bmatrix}
-{}^{i}_{j}\mathrm{R}^\top \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix} \\
\operatorname{atan2}\big({}^{i}_{j}\mathrm{R}_{3,1}\sin\psi_{ji} - {}^{i}_{j}\mathrm{R}_{3,2}\cos\psi_{ji},\;\; -{}^{i}_{j}\mathrm{R}_{2,1}\sin\psi_{ji} + {}^{i}_{j}\mathrm{R}_{2,2}\cos\psi_{ji}\big) \\
\operatorname{atan2}\big(-{}^{i}_{j}\mathrm{R}_{1,3},\;\; {}^{i}_{j}\mathrm{R}_{1,1}\cos\psi_{ji} + {}^{i}_{j}\mathrm{R}_{1,2}\sin\psi_{ji}\big) \\
\operatorname{atan2}\big({}^{i}_{j}\mathrm{R}_{1,2},\;\; {}^{i}_{j}\mathrm{R}_{1,1}\big)
\end{bmatrix}
\end{aligned}$$

with Jacobian

$$\mathrm{J}_{\ominus} = \frac{\partial\mathbf{x}_{ji}}{\partial\mathbf{x}_{ij}} = \begin{bmatrix} -{}^{i}_{j}\mathrm{R}^\top & \mathrm{N} \\ 0_{3\times3} & \mathrm{Q} \end{bmatrix}$$

where

$$\mathrm{N} = \begin{bmatrix}
0 & -{}^{i}_{j}\mathrm{R}_{3,1}\big(x_{ij}\cos\psi_{ij} + y_{ij}\sin\psi_{ij}\big) + z_{ij}\cos\theta_{ij} & {}^{i}_{j}\mathrm{R}_{2,1}x_{ij} - {}^{i}_{j}\mathrm{R}_{1,1}y_{ij} \\
z_{ji} & -{}^{i}_{j}\mathrm{R}_{3,2}\big(x_{ij}\cos\psi_{ij} + y_{ij}\sin\psi_{ij}\big) + z_{ij}\sin\theta_{ij}\sin\phi_{ij} & {}^{i}_{j}\mathrm{R}_{2,2}x_{ij} - {}^{i}_{j}\mathrm{R}_{1,2}y_{ij} \\
-y_{ji} & -{}^{i}_{j}\mathrm{R}_{3,3}\big(x_{ij}\cos\psi_{ij} + y_{ij}\sin\psi_{ij}\big) + z_{ij}\sin\theta_{ij}\cos\phi_{ij} & {}^{i}_{j}\mathrm{R}_{2,3}x_{ij} - {}^{i}_{j}\mathrm{R}_{1,3}y_{ij}
\end{bmatrix}$$

$$\mathrm{Q} = \frac{1}{(1 - {}^{i}_{j}\mathrm{R}^2_{1,3})}\begin{bmatrix}
-{}^{i}_{j}\mathrm{R}_{1,1} & -{}^{i}_{j}\mathrm{R}_{1,2}\cos\phi_{ij} & {}^{i}_{j}\mathrm{R}_{1,3}{}^{i}_{j}\mathrm{R}_{3,3} \\
{}^{i}_{j}\mathrm{R}_{1,2}\sqrt{(1 - {}^{i}_{j}\mathrm{R}^2_{1,3})} & -{}^{i}_{j}\mathrm{R}_{3,3}\cos\psi_{ij}\sqrt{(1 - {}^{i}_{j}\mathrm{R}^2_{1,3})} & {}^{i}_{j}\mathrm{R}_{2,3}\sqrt{(1 - {}^{i}_{j}\mathrm{R}^2_{1,3})} \\
{}^{i}_{j}\mathrm{R}_{1,3}{}^{i}_{j}\mathrm{R}_{1,1} & -{}^{i}_{j}\mathrm{R}_{2,3}\cos\psi_{ij} & -{}^{i}_{j}\mathrm{R}_{3,3}
\end{bmatrix}.$$

Example:
Given the vehicle pose in the local-level frame, $\mathbf{x}_{\ell v}$, we can express the local-level frame from the vehicle's point of view as

$$\mathbf{x}_{v\ell} = \ominus\mathbf{x}_{\ell v}.$$

Assuming that vehicle pose $\mathbf{x}_{\ell v}$ is a random variable with covariance $\Sigma_{\mathbf{x}_{\ell v}}$, then to first-order the covariance of the local-level frame with respect to the vehicle is

$$\Sigma_{\mathbf{x}_{v\ell}} = \mathrm{J}_{\ominus}\Sigma_{\mathbf{x}_{\ell v}}\mathrm{J}_{\ominus}^\top.$$

Example A.2 Using the inverse operation to express local-level with respect to the vehicle frame.

A.2.4 Tail-to-Tail Operation

Finally, we come to the tail-to-tail operation which is a *composite* relationship built upon the head-to-tail and inverse operation. This composite operation occurs frequently and therefore is worthwhile to define on its own. The tail-to-tail operation is used to express the relative-pose between two frames that are represented in a common coordinate system. For example, given pose vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{ik}$, the tail-to-tail operation yields the relative-pose $\mathbf{x}_{jk}$. Using standard homogeneous coordinate frame notation we can derive this composite operation as

$$ {}^{j}_{k}\mathrm{H} = {}^{j}_{i}\mathrm{H}\, {}^{i}_{k}\mathrm{H} = {}^{i}_{j}\mathrm{H}^{-1}\, {}^{i}_{k}\mathrm{H}. $$

Similarly, using the head-to-tail and inverse relationship we can equivalently define $\mathbf{x}_{jk}$ directly in component form as

$$ \begin{aligned} \mathbf{x}_{jk} &= \mathbf{x}_{ji} \oplus \mathbf{x}_{ik} \\ &= \ominus\mathbf{x}_{ij} \oplus \mathbf{x}_{ik}. \end{aligned} $$

The associated Jacobian of this composite operation can be obtained by chain-rule as

$$ \begin{aligned} {}_{\ominus}\mathrm{J}_{\oplus} &= \frac{\partial \mathbf{x}_{jk}}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{ik})} \\ &= \frac{\partial \mathbf{x}_{jk}}{\partial(\mathbf{x}_{ji}, \mathbf{x}_{ik})} \cdot \frac{\partial(\mathbf{x}_{ji}, \mathbf{x}_{ik})}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{ik})} \\ &= \mathrm{J}_{\oplus} \cdot \begin{bmatrix} \mathrm{J}_{\ominus} & 0_{6\times6} \\ 0_{6\times6} & \mathrm{I}_{6\times6} \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{J}_{\oplus1}\mathrm{J}_{\ominus} & \mathrm{J}_{\oplus2} \end{bmatrix}. \end{aligned} $$

Example:
Given local-level vehicle poses $\mathbf{x}_{\ell v_i}$ and $\mathbf{x}_{\ell v_j}$ corresponding to times t_i and t_j respectively, we can express their relative pose $\mathbf{x}_{v_i v_j}$ as

$$ \mathbf{x}_{v_i v_j} = \ominus\mathbf{x}_{\ell v_i} \oplus \mathbf{x}_{\ell v_j}. $$

Assuming the two poses $\mathbf{x}_{\ell v_i}$ and $\mathbf{x}_{\ell v_j}$ are random variables with covariances $\Sigma_{\mathbf{x}_{\ell v_i}}$ and $\Sigma_{\mathbf{x}_{\ell v_j}}$ and cross-covariance $\Sigma_{\mathbf{x}_{\ell v_i}\mathbf{x}_{\ell v_i}}$, then to first-order the covariance of their relative relationship is

$$ \Sigma_{\mathbf{x}_{v_i v_j}} = {}_{\ominus}\mathrm{J}_{\oplus} \begin{bmatrix} \Sigma_{\mathbf{x}_{\ell v_i}} & \Sigma_{\mathbf{x}_{\ell v_i}\mathbf{x}_{\ell v_j}} \\ \Sigma^{\top}_{\mathbf{x}_{\ell v_i}\mathbf{x}_{\ell v_j}} & \Sigma_{\mathbf{x}_{\ell v_j}} \end{bmatrix} {}_{\ominus}\mathrm{J}^{\top}_{\oplus}. $$

Example A.3 Using the tail-to-tail operation to compute the relative-pose between two time-sampled vehicle poses.

A.3 Vehicle Model

A vehicle model is a useful mathematical concept used to describe the change in vehicle state in response to control inputs (or lack thereof) and is often a required component in a general sensor fusion framework that tries to estimate the vehicle's state based upon noisy sensor measurements; see any of [7, 18, 54] for reference. We choose to model the vehicle dynamics using a 6-DOF constant velocity process model. This model sufficiently captures the characteristically slow-dynamics of most underwater imaging platforms used in a structured survey context and, additionally, can be applied across multiple vehicle platforms without reformulation.[5]

A.3.1 6-DOF Continuous-Time Constant Velocity Process Model

Deterministic Description

The vehicle state $\mathbf{x}_v$ is defined by the 12-vector:

$$\mathbf{x}_v = \left[{}^{\ell}\mathbf{t}_{\ell v}^{\top},\quad \mathbf{\Theta}_{\ell v}^{\top},\quad {}^{v}\boldsymbol{\nu}^{\top},\quad {}^{v}\boldsymbol{\omega}^{\top}\right]^{\top} \tag{A.1}$$

where $\mathbf{x}_{\ell v} = \left[{}^{\ell}\mathbf{t}_{\ell v}{}^{\top}, \mathbf{\Theta}_{\ell v}^{\top}\right]^{\top}$ is the local-level vehicle pose as defined in §A.2.1, ${}^{v}\boldsymbol{\nu} = [u, v, w]$ are the body-frame linear velocities, and ${}^{v}\boldsymbol{\omega} = [p, q, r]$ are the body-frame angular rates. Under the constant velocity approximation, this state description allows us to define the deterministic component of the continuous-time motion-model as

$$\dot{\mathbf{x}}_v(t) = \mathbf{f}(\mathbf{x}_v(t)) \tag{A.2a}$$

$$\frac{d}{dt}\begin{bmatrix} {}^{\ell}\mathbf{t}_{\ell v} \\ \mathbf{\Theta}_{\ell v} \\ {}^{v}\boldsymbol{\nu} \\ {}^{v}\boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} {}^{\ell}_{v}\mathrm{R}\,{}^{v}\boldsymbol{\nu} \\ \mathcal{J}\,{}^{v}\boldsymbol{\omega} \\ 0_{3\times 1} \\ 0_{3\times 1} \end{bmatrix} \tag{A.2b}$$

where ${}^{\ell}_{v}\mathrm{R}$ is the orthonormal rotation matrix rotating the body-frame velocities into the local-level frame and $\mathcal{J}$ is a 3 × 3 matrix mapping body-frame rates to Euler rates. Note that both ${}^{\ell}_{v}\mathrm{R}$ and $\mathcal{J}$ have a nonlinear dependence on $\mathbf{\Theta}_{\ell v}$. The mapping $\mathcal{J}$ can be derived from first-principles by considering the inverse relationship whereby the principle rotation sequence $\mathrm{rotz}(\psi) \rightarrow \mathrm{roty}(\theta) \rightarrow \mathrm{rotx}(\phi)$ (see §A.2.1) is used to map Euler rates to body rates as

$$\begin{aligned}\begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathrm{rotx}(\phi)\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathrm{rotx}(\phi)\,\mathrm{roty}(\theta)\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}}_{\mathcal{J}^{-1}} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}.\end{aligned}$$

[5] The process noise autocorrelation matrix Q in (A.3) may need to be appropriately tuned.

Thus, the mapping from body-frame rates to Euler rates is given by

$$\mathcal{J} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}.$$

In §A.3.2 the process-model Jacobian will be required, so for completeness we present it here. To derive the Jacobian we first define the following quantities:

$$\mathrm{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad \mathrm{R}_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad \mathrm{R}_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\dot{\mathrm{R}}_\phi = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin\phi & \cos\phi \\ 0 & -\cos\phi & -\sin\phi \end{bmatrix} \quad \dot{\mathrm{R}}_\theta = \begin{bmatrix} -\sin\theta & 0 & -\cos\theta \\ 0 & 0 & 0 \\ \cos\theta & 0 & -\sin\theta \end{bmatrix} \quad \dot{\mathrm{R}}_\psi = \begin{bmatrix} -\sin\psi & \cos\psi & 0 \\ -\cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\dot{\mathcal{J}}_\phi = \begin{bmatrix} 0 & \cos\phi\tan\theta & -\sin\phi\tan\theta \\ 0 & -\sin\phi & -\cos\phi \\ 0 & \cos\phi\sec\theta & -\sin\phi\sec\theta \end{bmatrix} \quad \dot{\mathcal{J}}_\theta = \begin{bmatrix} 0 & \sin\phi\sec^2\theta & \cos\phi\sec^2\theta \\ 0 & 0 & 0 \\ 0 & \sin\phi\sec\theta\tan\theta & \cos\phi\sec\theta\tan\theta \end{bmatrix}$$

Using the above definitions the nonzero partials of the process-model Jacobian are:

$$\frac{\partial {}^\ell\dot{\mathbf{t}}_{\ell v}}{\partial\phi} = \mathrm{R}_\psi^\top \mathrm{R}_\theta^\top \dot{\mathrm{R}}_\phi^\top {}^v\boldsymbol{\nu} \quad \frac{\partial {}^\ell\dot{\mathbf{t}}_{\ell v}}{\partial\theta} = \mathrm{R}_\psi^\top \dot{\mathrm{R}}_\theta^\top \mathrm{R}_\phi^\top {}^v\boldsymbol{\nu} \quad \frac{\partial {}^\ell\dot{\mathbf{t}}_{\ell v}}{\partial\psi} = \dot{\mathrm{R}}_\psi^\top \mathrm{R}_\theta^\top \mathrm{R}_\phi^\top {}^v\boldsymbol{\nu} \quad \frac{\partial {}^\ell\dot{\mathbf{t}}_{\ell v}}{\partial {}^v\boldsymbol{\nu}} = {}^\ell_v\mathrm{R}$$

$$\frac{\partial \dot{\boldsymbol{\Theta}}_{\ell v}}{\partial\phi} = \dot{\mathcal{J}}_\phi {}^v\boldsymbol{\omega} \quad \frac{\partial \dot{\boldsymbol{\Theta}}_{\ell v}}{\partial\theta} = \dot{\mathcal{J}}_\theta {}^v\boldsymbol{\omega} \quad \frac{\partial \dot{\boldsymbol{\Theta}}_{\ell v}}{\partial\psi} = 0_{3\times1} \quad \frac{\partial \dot{\boldsymbol{\Theta}}_{\ell v}}{\partial {}^v\boldsymbol{\omega}} = \mathcal{J}$$

Hence, the total process model Jacobian is given by

$$\mathrm{F}_\mathrm{x} = \frac{\partial \mathbf{f}(\mathbf{x}_v)}{\partial \mathbf{x}_v} = \begin{bmatrix} 0_{3\times3} & \frac{\partial {}^\ell\dot{\mathbf{t}}_{\ell v}}{\partial \boldsymbol{\Theta}_{\ell v}} & \frac{\partial {}^\ell\dot{\mathbf{t}}_{\ell v}}{\partial {}^v\boldsymbol{\nu}} & 0_{3\times3} \\ 0_{3\times3} & \frac{\partial \dot{\boldsymbol{\Theta}}_{\ell v}}{\partial \boldsymbol{\Theta}_{\ell v}} & 0_{3\times3} & \frac{\partial \dot{\boldsymbol{\Theta}}_{\ell v}}{\partial {}^v\boldsymbol{\omega}} \\ 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & 0_{6\times3} \end{bmatrix}.$$

Probabilistic Description

Clearly, the deterministic motion-model of (A.2) represents an approximation to the true vehicle dynamics. As is typical engineering practice, we account for this modeling error by augmenting our motion-model to include a probabilistic term which reflects our negligence of model detail:

$$\dot{\mathbf{x}}_v(t) = \mathbf{f}(\mathbf{x}_v(t)) + \mathrm{G}\mathbf{w}(t). \tag{A.3}$$

Here, $\mathrm{G} = \left[\begin{smallmatrix} 0_{6\times6} \\ \mathrm{I}_{6\times6} \end{smallmatrix}\right]$ is a gain matrix mapping the 6-vector zero-mean white Gaussian noise process $\mathbf{w}(t)$ with covariance $E[\mathbf{w}(t)\mathbf{w}(\tau)^\top] = \mathrm{Q}\delta(t-\tau)$ to the rate derivatives ${}^v\dot{\boldsymbol{\nu}}$ and ${}^v\dot{\boldsymbol{\omega}}$. This white process noise reflects the fact that the rates ${}^v\boldsymbol{\nu}$ and ${}^v\boldsymbol{\omega}$ of (A.2) are not truly constant, but instead are approximated by a random walk variation in time.

A.3.2 Continuous-Time to Discrete-Time Mapping

While the continuous-time formulation of (A.3) is the most natural representation for analytically expressing the vehicle dynamics, it must be translated from its native formulation into a discrete-time representation if we are to implement it within a computer algorithm. One straightforward technique (which conveniently requires zero-reformulation) is to use a fourth-order Runge-Kutta numerical integration strategy [119] to propagate the vehicle state forward in between discrete samples in time (e.g., this technique can be used within a continuous-discrete Kalman filter formulation [54] to advance the vehicle state between asynchronous sensor measurements as described in Chapter 2). However, for the information framework presented later in Chapter 4 it will prove useful to define the vehicle model directly as a discrete-time difference equation.

Piecewise-Constant Discrete-Time Difference Equation

Suppose that at time t_k we have an estimate for $\mathbf{x}_v(t_k)$ denoted $\boldsymbol{\mu}_{t_k}$ — expanding (A.3) in a Taylor series about this point, we get

$$\dot{\mathbf{x}}_v(t) = \mathbf{f}(\boldsymbol{\mu}_{t_k}) + \mathrm{F}_\mathbf{x}\left(\mathbf{x}_v(t) - \boldsymbol{\mu}_{t_k}\right) + \mathsf{HOT} + \mathrm{G}\mathbf{w}(t).$$

Here, $\mathrm{F}_\mathbf{x} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_v)}{\partial \mathbf{x}_v}\right|_{\mathbf{x}_v = \boldsymbol{\mu}_{t_k}}$ is the process model Jacobian and HOT denotes higher-order terms in the expansion. Dropping the higher-order terms and rearranging we have

$$\begin{aligned}\dot{\mathbf{x}}_v(t) &\approx \mathrm{F}_\mathbf{x}\mathbf{x}_v(t) + \underbrace{\left(\mathbf{f}(\boldsymbol{\mu}_{t_k}) - \mathrm{F}_\mathbf{x}\boldsymbol{\mu}_{t_k}\right)}_{\mathbf{u}(t_k)} + \mathrm{G}\mathbf{w}(t) \\ &= \mathrm{F}_\mathbf{x}\mathbf{x}_v(t) + \mathbf{u}(t_k) + \mathrm{G}\mathbf{w}(t)\end{aligned}$$

where $\left(\mathbf{f}(\boldsymbol{\mu}_{t_k}) - \mathrm{F}_\mathbf{x}\boldsymbol{\mu}_{t_k}\right)$ masquerades as a constant input pseudo control $\mathbf{u}(t_k)$.

Assuming that $\mathbf{u}(t_k)$ and $\mathrm{F}_\mathbf{x}$ remain constant over a short time step $\Delta t = [t_k, t_{k+1})$, we can piecewise approximate our nonlinear partial differential equation (PDE) (A.3) over the time interval Δt with the constant coefficient PDE specified in (A.4)

$$\dot{\mathbf{x}}_v(t) = \mathrm{F}_\mathbf{x}\mathbf{x}_v(t) + \mathrm{B}\mathbf{u}(t) + \mathrm{G}\mathbf{w}(t) \tag{A.4}$$

where B is simply the identity matrix and $\mathbf{u}(t) = \mathbf{u}(t_k)$ for $t_k \leq t < t_{k+1}$. Now that we have a linear constant coefficient PDE, we can map this directly to a discrete-time difference equation sampled at arbitrary times [7]:

$$\mathbf{x}_v[t_{k+1}] = \mathrm{F}_k\mathbf{x}_v[t_k] + \mathrm{B}_k\mathbf{u}[t_k] + \mathbf{w}[t_k]. \tag{A.5}$$

To evaluate the parameters of (A.5) we note the following. For the constant coefficient PDE of (A.4) the corresponding discrete-time state transition matrix F_k is given by

$$\mathrm{F}_k = e^{\mathrm{F}_\mathbf{x}\Delta t}.$$

The discrete-time control gain B_k is computed as[6]

$$\mathrm{B}_k = \int_{t_k}^{t_{k+1}} e^{\mathrm{F}_\mathrm{x}(t_{k+1}-\tau)}\mathrm{B}d\tau = \int_{t_k}^{t_{k+1}} e^{\mathrm{F}_\mathrm{x}(t_{k+1}-\tau)}d\tau = e^{\mathrm{F}_\mathrm{x}t_{k+1}}\int_{t_k}^{t_{k+1}} e^{-\mathrm{F}_\mathrm{x}\tau}d\tau.$$

And finally, the zero-mean discrete-time white Gaussian process noise $\mathbf{w}[t_k]$ is related to the zero-mean continuous-time white Gaussian noise $\mathbf{w}(t)$ by

$$\mathbf{w}[t_k] = \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau)}\mathrm{G}\mathbf{w}(\tau)d\tau$$

with covariance

$$\mathrm{Q}_k = E\Big[\mathbf{w}[t_k]\mathbf{w}[t_k]^\top\Big] = \int_{t_k}^{t_{k+1}} e^{\mathrm{F}_\mathrm{x}(t_{k+1}-\tau)}\mathrm{G}\mathrm{Q}(\tau)\mathrm{G}^\top e^{\mathrm{F}_\mathrm{x}^\top(t_{k+1}-\tau)}d\tau.$$

For the fixed parameter case of (A.4) the evaluation of Q_k simplifies considerably and can be computed using Van Loan's method [18, §5.3]. In summary, Algorithm A.1 describes the continuous to discrete mapping as we implement it.

A.4 Sensor Observation Models

Using our description of vehicle state defined in (A.1), in this section we describe our models for strap-down sensor measured quantities.[7] The main idea behind an observation model is to use our knowledge of vehicle state to try and "predict" sensor observed quantities in the sensor frame of reference — the reason for choosing the sensor frame of reference is because this is where the measurement "noise" is most naturally expressed. Within a typical state estimator framework, the discrepancy between predicted and measured quantities is used in a weighted update to modify our belief in what the sensor is telling us versus what our kinematics (i.e., process-model) tell us.

Note that in deriving the following observation models we assume that the sensor to vehicle relative-pose quantity $\mathbf{x}_{vs} = [{}^v\mathbf{t}_{vs},\ \mathbf{\Theta}_{vs}]$ is known from calibration and that the vehicle state $\mathbf{x}_v$ is defined by (A.1).

A.4.1 DVL Observation Model

When operating in bottom-lock mode, the DVL measures seafloor-relative velocities as expressed in its sensor reference frame (refer to §1.3.2 for a more in-depth discussion). Accounting for the moment-arm between the body-fixed vehicle frame origin and the sensor

[6] Note that a closed-form solution does not exist because the Jacobian F_x is singular, therefore, we numerically evaluate the definite integral using a sufficient number of intervals and Simpson's Rule [119].

[7] Camera derived quantities are discussed in Chapter 2 where we introduce the concept of visually augmented navigation.

Algorithm A.1 Piecewise-constant continuous-time to discrete-time mapping.

```
for time step $t_k$ do
    Evaluate the continuous-time process-model (A.3) about the linearization point $\boldsymbol{\mu}_{t_k}$ to get $\mathbf{f}(\boldsymbol{\mu}_{t_k})$ and the Jacobian $\mathrm{F}_{\mathrm{x}}$.
    Form the block-matrix X.
        $\mathrm{X} = \left[\begin{array}{c|c} -\mathrm{F}_{\mathrm{x}} & \mathrm{GQG}^{\top} \\ \hline 0 & \mathrm{F}_{\mathrm{x}}^{\top} \end{array}\right]$
    Evaluate the matrix exponential and call it Y.
        $\mathrm{Y} = e^{\mathrm{X}\Delta t} = \left[\begin{array}{c|c} \cdots & \mathrm{F}_k^{-1}\mathrm{Q}_k \\ \hline 0 & \mathrm{F}_k^{\top} \end{array}\right]$
    $\mathrm{F}_k \leftarrow$ (lower-right block of Y)$^{\top}$
    $\mathrm{Q}_k \leftarrow \mathrm{F}_k \times$ (upper-right block of Y)
    $\mathbf{u}[t_k] \leftarrow \mathbf{f}(\boldsymbol{\mu}_{t_k}) - \mathrm{F}_{\mathrm{x}}\boldsymbol{\mu}_{t_k}$
    $\mathrm{B}_k \leftarrow e^{\mathrm{F}_{\mathrm{x}} t_{k+1}} \int_{t_k}^{t_{k+1}} e^{-\mathrm{F}_{\mathrm{x}}\tau} d\tau$    numerically evaluate over a sufficient number of intervals using Simpson's Rule
    Evaluate the discrete-time model (A.5) for the current set of parameters.
end for
```

frame we have [70]

$$\begin{aligned}\mathbf{v}'_{\mathsf{sensor}} &= {}^{s}_{v}\mathrm{R}\big({}^{v}\boldsymbol{\nu} + {}^{v}\boldsymbol{\omega} \times {}^{v}\mathbf{t}_{vs}\big)\\ &= {}^{s}_{v}\mathrm{R}\big({}^{v}\boldsymbol{\nu} - [{}^{v}\mathbf{t}_{vs}]_{\times}{}^{v}\boldsymbol{\omega}\big)\end{aligned}$$

where the $[.]_{\times}$ operator represents a skew-symmetric matrix implementing the vector cross-product.[8] Therefore, the linear state observation model is given by[9]

$$\begin{aligned}\mathbf{z} &= \mathrm{H}\mathbf{x}_v\\ &= \begin{bmatrix}0_{3\times3} & 0_{3\times3} & {}^{s}_{v}\mathrm{R} & -{}^{s}_{v}\mathrm{R}[{}^{v}\mathbf{t}_{vs}]_{\times}\end{bmatrix}\mathbf{x}_v.\end{aligned}$$

A.4.2 Angular Rate Sensor Observation Models

Since the sensor and vehicle can be considered as a rigid-body, the sensor measured angular rates are simply the vehicle angular rates rotated into the sensor coordinate frame:

$$\boldsymbol{\omega}_{\mathsf{sensor}} = {}^{s}_{v}\mathrm{R}^{v}\boldsymbol{\omega}.$$

Therefore, the linear state observation model is given by

$$\begin{aligned}\mathbf{z} &= \mathrm{H}\mathbf{x}_v\\ &= \begin{bmatrix}0_{3\times3} & 0_{3\times3} & 0_{3\times3} & {}^{s}_{v}\mathrm{R}\end{bmatrix}\mathbf{x}_v.\end{aligned}$$

A.4.3 Attitude Sensor Observation Model

For the DVL and angular rate sensor, the rate measurements they provide are described in a sensor relative coordinate-frame that does not depend upon the definition of an external reference frame. However, in the case of absolute orientation measurements as measured by an attitude module consisting of a compass and tilt sensors, its definition of orientation is with respect to a particular reference frame. Therefore, when deriving the attitude sensor observation model we must consider that the sensor's external reference frame, denoted subscript r, may not coincide with the local-level definition used by the vehicle. Hence, the static pose of local-level with respect to the sensor reference frame (i.e., $\mathbf{x}_{r\ell}$) will have to be considered in the general case.

The predicted sensor pose is given by

$$\mathbf{x}_{rs} = \mathbf{x}_{r\ell} \oplus (\mathbf{x}_{\ell v} \oplus \mathbf{x}_{vs})$$

[8] The skew-symmetric matrix S for the 3-vector $\mathbf{s} = [s_1, s_2, s_3]^\top$ is given by

$$\mathrm{S} = [\mathbf{s}]_{\times} = \begin{bmatrix}0 & -s_3 & s_2\\ s_3 & 0 & -s_1\\ -s_2 & s_1 & 0\end{bmatrix}.$$

[9] We model the DVL as providing a 3-vector measurement of Euclidean velocities as measured in the sensor frame when in actuality it measures velocity components along each of its 4 acoustic beams. Unfortunately, our data logging strategy only records the resolved 3-vector sensor velocities which prevents us from modeling beam-level detail. However, note that to model the actual beam velocity measurements the only modification that is required is to premultiply H with the static beam-geometry transformation matrix T [131].

with relevant partial derivative

$$\frac{\partial \mathbf{x}_{rs}}{\partial \mathbf{x}_{\ell v}} = \frac{\partial \mathbf{x}_{rs}}{\partial \mathbf{x}_{\ell s}} \cdot \frac{\partial \mathbf{x}_{\ell s}}{\partial \mathbf{x}_{\ell v}} = \mathrm{J}_{\oplus 2}\Big|_{(\mathbf{x}_{r\ell},\ \mathbf{x}_{\ell s})} \cdot \mathrm{J}_{\oplus 1}\Big|_{(\mathbf{x}_{\ell v},\ \mathbf{x}_{vs})}.$$

However, we are only interested in the attitude portion which can be extracted as

$$\mathbf{\Theta}_{\mathsf{sensor}} = \underbrace{\begin{bmatrix} 0_{3\times 3} & \mathrm{I}_{3\times 3} \end{bmatrix}}_{\mathrm{A}} \mathbf{x}_{rs}.$$

Therefore, the nonlinear observation model is given by

$$\begin{aligned} \mathbf{z} &= \mathbf{h}(\mathbf{x}_v) \\ &= \mathrm{A}\big(\mathbf{x}_{r\ell} \oplus (\mathbf{x}_{\ell v} \oplus \mathbf{x}_{vs})\big) \end{aligned}$$

with Jacobian

$$\mathrm{H}_{\mathsf{x}} = \mathrm{A}\big(\mathrm{J}_{\oplus 2}\Big|_{(\mathbf{x}_{r\ell},\ \mathbf{x}_{\ell s})} \cdot \mathrm{J}_{\oplus 1}\Big|_{(\mathbf{x}_{\ell v},\ \mathbf{x}_{vs})}\big).$$

A.4.4 Depth Sensor Observation Model

For the depth sensor, the predicted sensor pose is given by

$$\mathbf{x}_{\ell s} = \mathbf{x}_{\ell v} \oplus \mathbf{x}_{vs}$$

of which we are only interested in the Z-component which can be extracted as

$$z_{\mathsf{sensor}} = \underbrace{[0, 0, 1, 0, 0, 0]}_{\mathrm{A}} \mathbf{x}_{\ell s}.$$

Therefore, the scalar nonlinear observation model is given by

$$\begin{aligned} z &= h(\mathbf{x}_v) \\ &= \mathrm{A}\big(\mathbf{x}_{\ell v} \oplus \mathbf{x}_{vs}\big) \end{aligned}$$

with Jacobian

$$\mathrm{H}_{\mathsf{x}} = \mathrm{A}\mathrm{J}_{\oplus 1}.$$

APPENDIX B

Accompanying Derivations for SEIFs

In this chapter we derive from first principles some of the expressions of the Gaussian information form that were given in Chapter 3. In particular, we derive the expressions for marginalization and conditioning in the information form, as well as the expressions for the both the original and modified SEIF sparsification rules.

B.1 Mechanics of the Information Form

Suppose the normal random variable $\boldsymbol{\xi}$ is written in partitioned form as $\boldsymbol{\xi} = [\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top]^\top$ where

$$
\begin{aligned}
p(\boldsymbol{\xi}) &= \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}^{-1}(\boldsymbol{\eta}, \Lambda) \\
\Rightarrow p(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{bmatrix}, \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}\Big) = \mathcal{N}^{-1}\Big(\begin{bmatrix} \boldsymbol{\eta}_\alpha \\ \boldsymbol{\eta}_\beta \end{bmatrix}, \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}\Big)
\end{aligned}
$$

and by definition $\Lambda = \Sigma^{-1}$ and $\boldsymbol{\eta} = \Lambda\boldsymbol{\mu}$. In the following we derive the fundamental probabilistic operations of marginalization and conditioning in the information form.[1]

[1] The marginalization and conditioning expressions we derive hold in the general case for any partitioning of $\boldsymbol{\xi}$ since we can always reorder the elements into the form $\boldsymbol{\xi} = [\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top]^\top$ via an appropriate orthonormal permutation matrix A. For example, suppose $\boldsymbol{\xi} = [\mathbf{a}^\top, \mathbf{b}^\top, \mathbf{c}^\top]^\top$ with $p(\boldsymbol{\xi}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}^{-1}(\boldsymbol{\eta}, \Lambda)$ and we want $\boldsymbol{\xi}' = [\mathbf{a}^\top, \mathbf{c}^\top, \mathbf{b}^\top]^\top$, then $\boldsymbol{\xi}' = \mathrm{A}\boldsymbol{\xi}$ where

$$
\mathrm{A} = \begin{bmatrix} \mathrm{I}_{a\times a} & 0_{a\times b} & 0_{a\times c} \\ 0_{c\times a} & 0_{c\times b} & \mathrm{I}_{c\times c} \\ 0_{b\times a} & \mathrm{I}_{b\times b} & 0_{b\times c} \end{bmatrix}
$$

and since $\boldsymbol{\xi}'$ is a linear transformation of $\boldsymbol{\xi}$, it remains a normal random variable with statistics

$$
\begin{aligned}
\boldsymbol{\mu}' &= \mathrm{A}\boldsymbol{\mu} \\
\Sigma' &= \mathrm{A}\Sigma\mathrm{A}^\top
\end{aligned}
\quad \text{and} \quad
\begin{aligned}
\boldsymbol{\eta}' &= \Lambda'\boldsymbol{\mu}' = \mathrm{A}\Lambda\mathrm{A}^\top\mathrm{A}\boldsymbol{\mu} = \mathrm{A}\boldsymbol{\eta} \\
\Lambda' &= (\Sigma')^{-1} = \mathrm{A}^{-\top}\Sigma^{-1}\mathrm{A}^{-1} = \mathrm{A}\Lambda\mathrm{A}^\top
\end{aligned}
$$

B.1.1 Marginalization

Suppose we want the distribution over $\boldsymbol{\alpha}$ only, then to obtain $p(\boldsymbol{\alpha})$ we must marginalize out $\boldsymbol{\beta}$'s cumulative effect via integration:

$$p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\beta} = \int \mathcal{N}(\boldsymbol{\mu}, \Sigma) d\boldsymbol{\beta} = \int \mathcal{N}^{-1}(\boldsymbol{\eta}, \Lambda) d\boldsymbol{\beta}.$$

Covariance Form

In covariance form, rather than actually computing the indefinite integral over $\boldsymbol{\beta}$, we can obtain the same result by considering the following linear transformation:

$$\mathbf{z} = \boldsymbol{\alpha} = \mathrm{A}\boldsymbol{\xi} \quad \text{where} \quad \mathrm{A} = \begin{bmatrix} \mathrm{I}_{\alpha\times\alpha} & 0_{\alpha\times\beta} \end{bmatrix},$$

which implies that the mean and covariance of the normal random variable $\mathbf{z}$ are given by

$$\begin{aligned} \boldsymbol{\mu}_z &= \mathrm{A}\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha \\ \Sigma_{zz} &= \mathrm{A}\Sigma\mathrm{A}^\top = \Sigma_{\alpha\alpha}. \end{aligned} \tag{B.1}$$

Hence, marginalization is a constant-time operation in covariance form, because we simply extract the appropriate sub-elements/sub-block from the mean-vector/covariance-matrix respectively.

Inversion of a Partitioned Matrix

Before we derive the equivalent operation in the information form, a useful result from linear algebra that we'll employ is the inversion of a nonsingular block 2×2 matrix which we repeat here for convenience. Quoting Bar-Shalom [7, §1.3.3]:

> The inverse of the (nonsingular) $n \times n$ ***partitioned matrix***
>
> $$\begin{bmatrix} \mathrm{P}_{11} & \mathrm{P}_{12} \\ \mathrm{P}_{21} & \mathrm{P}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathrm{V}_{11} & \mathrm{V}_{12} \\ \mathrm{V}_{21} & \mathrm{V}_{22} \end{bmatrix}$$
>
> where P_{11} is $n_1 \times n_1$, P_{12} is $n_1 \times n_2$, P_{21} is $n_2 \times n_1$, P_{22} is $n_2 \times n_2$ and $n_1 + n_2 = n$, has the partitions
>
> $$\mathrm{V}_{11} = \mathrm{P}_{11}^{-1} + \mathrm{P}_{11}^{-1}\mathrm{P}_{12}\mathrm{V}_{22}\mathrm{P}_{21}\mathrm{P}_{11}^{-1} = (\mathrm{P}_{11} - \mathrm{P}_{12}\mathrm{P}_{22}^{-1}\mathrm{P}_{21})^{-1} \tag{B.2}$$
> $$\mathrm{V}_{12} = -\mathrm{P}_{11}^{-1}\mathrm{P}_{12}\mathrm{V}_{22} = -\mathrm{V}_{11}\mathrm{P}_{12}\mathrm{P}_{22}^{-1} \tag{B.3}$$
> $$\mathrm{V}_{21} = -\mathrm{V}_{22}\mathrm{P}_{21}\mathrm{P}_{11}^{-1} = -\mathrm{P}_{22}^{-1}\mathrm{P}_{21}\mathrm{V}_{11} \tag{B.4}$$
> $$\mathrm{V}_{22} = \mathrm{P}_{22}^{-1} + \mathrm{P}_{22}^{-1}\mathrm{P}_{21}\mathrm{V}_{11}\mathrm{P}_{12}\mathrm{P}_{22}^{-1} = (\mathrm{P}_{22} - \mathrm{P}_{21}\mathrm{P}_{11}^{-1}\mathrm{P}_{12})^{-1} \tag{B.5}$$

Information Form

To obtain the expression for marginalization in the information form we simply transform the covariance form result of (B.1) according to

$$\begin{aligned} \Lambda_{zz} &= \Sigma_{zz}^{-1} \\ &= \Sigma_{\alpha\alpha}^{-1} \\ &\overset{(B.2)}{=} \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha} \end{aligned} \qquad \begin{aligned} \boldsymbol{\eta}_z &= \Lambda_{zz}\boldsymbol{\mu}_z \\ &= \big(\Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}\big)\boldsymbol{\mu}_\alpha \\ &= \Lambda_{\alpha\alpha}\boldsymbol{\mu}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}\boldsymbol{\mu}_\alpha \\ &= (\boldsymbol{\eta}_\alpha - \Lambda_{\beta\alpha}\boldsymbol{\mu}_\beta) - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}(\boldsymbol{\eta}_\beta - \Lambda_{\beta\beta}\boldsymbol{\mu}_\beta) \\ &= \boldsymbol{\eta}_\alpha - \Lambda_{\beta\alpha}\boldsymbol{\mu}_\beta - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta + \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\beta}\boldsymbol{\mu}_\beta \\ &= \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta. \end{aligned} \tag{B.6}$$

Hence, while marginalization is a constant-time operation in covariance form, it is minimally a cubic operation in the size of $\boldsymbol{\beta}$ in the canonical form due to the inversion of $\Lambda_{\beta\beta}$ in (B.6).

B.1.2 Conditioning

Suppose we want the distribution over $\boldsymbol{\alpha}$ conditioned on the random variable $\boldsymbol{\beta}$, then to obtain $p(\boldsymbol{\alpha}|\boldsymbol{\beta})$ we must compute

$$p(\boldsymbol{\alpha}|\boldsymbol{\beta}) = \frac{p(\boldsymbol{\alpha},\boldsymbol{\beta})}{p(\boldsymbol{\beta})} = \frac{\mathcal{N}(\boldsymbol{\mu},\Sigma)}{\int \mathcal{N}(\boldsymbol{\mu},\Sigma)d\boldsymbol{\alpha}} = \frac{\mathcal{N}^{-1}(\boldsymbol{\eta},\Lambda)}{\int \mathcal{N}^{-1}(\boldsymbol{\eta},\Lambda)d\boldsymbol{\alpha}}.$$

Covariance Form

In covariance form, the expressions for the conditional mean and covariance are well-known and are given by [7]

$$\begin{aligned} \boldsymbol{\mu}_{\alpha|\beta} &= \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta) \\ \Sigma_{\alpha|\beta} &= \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}. \end{aligned} \tag{B.7}$$

Hence, while marginalization is a constant-time operation in covariance form, conditioning turns out to be minimally a cubic operation in the size of $\boldsymbol{\beta}$ due to the inversion of $\Sigma_{\beta\beta}$ in (B.7).

Information Form

To obtain the expressions for conditioning in the information form we simply transform the covariance form result of (B.7) according to

$$\begin{aligned} \Lambda_{\alpha|\beta} &= \Sigma_{\alpha|\beta}^{-1} \\ &= \big(\Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}\big)^{-1} \\ &\overset{(B.2)}{=} \Lambda_{\alpha\alpha} \end{aligned} \qquad \begin{aligned} \boldsymbol{\eta}_{\alpha|\beta} &= \Lambda_{\alpha|\beta}\boldsymbol{\mu}_{\alpha|\beta} \\ &= \Lambda_{\alpha\alpha}\big(\boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)\big) \\ &\overset{(B.3)}{=} \Lambda_{\alpha\alpha}\big(\boldsymbol{\mu}_\alpha - \Lambda_{\alpha\alpha}^{-1}\Lambda_{\alpha\beta}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)\big) \\ &= \big(\Lambda_{\alpha\alpha}\boldsymbol{\mu}_\alpha + \Lambda_{\alpha\beta}\boldsymbol{\mu}_\beta\big) - \Lambda_{\alpha\beta}\boldsymbol{\beta} \\ &= \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}. \end{aligned} \tag{B.8}$$

Hence, conditioning is a constant-time operation in the canonical form, because we simply extract the appropriate sub-elements/sub-block from the information-vector/information-matrix respectively.

B.2 SEIF Sparsification Rule

We begin by writing the sparsified posterior approximation as a ratio of the three individual distributions p_B, p_C, p_D as described in §3.3.1:

$$\tilde{p}_{\text{SEIF}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) = \frac{p_B(\mathbf{x}_t, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})}{p_C(\mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})} p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-). \tag{B.9}$$

Next, we calculate the individual terms of (B.9) by marginalizing and conditioning over our original distribution $p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) = \mathcal{N}(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)$.

B.2.1 Calculation of $p_A(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})$

This intermediate distribution will be used to compute both p_B and p_C and is obtained from our full posterior $p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$ by conditioning on the passive features with a realization of alpha (i.e., $\mathbf{m}^- = \boldsymbol{\alpha}$):

$$p_A(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha}) = \mathcal{N}(\mathrm{S}_{x_t,m^0,m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\mu}_A, \Sigma_A) = \mathcal{N}^{-1}(\mathrm{S}_{x_t,m^0,m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\eta}_A, \Lambda_A).$$

Covariance Form

$$\begin{aligned} \boldsymbol{\mu}_A &= \mathrm{S}_{x,m^0,m^+}^\top \boldsymbol{\mu}_t + \mathrm{S}_{x,m^0,m^+}^\top \overbrace{\Sigma_t \mathrm{S}_{m^-} \left(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-}\right)^{-1} \left(\boldsymbol{\alpha} - \mathrm{S}_{m^-}^\top \boldsymbol{\mu}_t\right)}^{\boldsymbol{\mu}_\alpha} \\ &= \mathrm{S}_{x,m^0,m^+}^\top \left(\boldsymbol{\mu}_t + \boldsymbol{\mu}_\alpha\right) \\ \Sigma_A &= \mathrm{S}_{x,m^0,m^+}^\top \Sigma_t \mathrm{S}_{x,m^0,m^+} - \mathrm{S}_{x,m^0,m^+}^\top \Sigma_t \mathrm{S}_{m^-} \left(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-}\right)^{-1} \mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{x,m^0,m^+} \end{aligned} \tag{B.10}$$

Information Form

$$\begin{aligned} \boldsymbol{\eta}_A &= \mathrm{S}_{x_t,m^0,m^+}^\top \boldsymbol{\eta}_t - \mathrm{S}_{x_t,m^0,m^+}^\top \overbrace{\Lambda_t \mathrm{S}_{m^-} \boldsymbol{\alpha}}^{\boldsymbol{\eta}_\alpha} \\ &= \mathrm{S}_{x_t,m^0,m^+}^\top \left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) \\ \Lambda_A &= \mathrm{S}_{x_t,m^0,m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^+} \end{aligned} \tag{B.11}$$

B.2.2 Calculation of $p_B(\mathbf{x}_t, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})$

This distribution is obtained by marginalizing out deactive features, $\mathbf{m}^0$, from p_A:

$$\begin{aligned} p_B(\mathbf{x}_t, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha}) &= \int p_A(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha}) d\mathbf{m}^0 \\ &= \mathcal{N}(\mathrm{S}_{x_t,m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\mu}_B, \Sigma_B) = \mathcal{N}^{-1}(\mathrm{S}_{x_t,m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\eta}_B, \Lambda_B). \end{aligned}$$

In the following, P denotes a projection matrix like S, but only for a subspace of $\boldsymbol{\xi}_t$.

Covariance Form

$$\begin{aligned}
\boldsymbol{\mu}_B &= \mathrm{P}_{xt,m^+}^\top \boldsymbol{\mu}_A \\
&= \mathrm{S}_{x,m^+}^\top \left(\boldsymbol{\mu}_t + \boldsymbol{\mu}_\alpha\right) \\
\Sigma_B &= \mathrm{P}_{xt,m^+}^\top \Sigma_A \mathrm{P}_{xt,m^+} \\
&= \mathrm{S}_{x,m^+}^\top \Sigma_t \mathrm{S}_{x,m^+} - \mathrm{S}_{x,m^+}^\top \Sigma_t \mathrm{S}_{m^-} \left(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-}\right)^{-1} \mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{x,m^+}
\end{aligned} \tag{B.12}$$

Information Form

$$\begin{aligned}
\boldsymbol{\eta}_B &= \mathrm{P}_{xt,m^+}^\top \boldsymbol{\eta}_A - \mathrm{P}_{xt,m^+}^\top \Lambda_A \mathrm{P}_{m^0} \left(\mathrm{P}_{m^0}^\top \Lambda_A \mathrm{P}_{m^0}\right)^{-1} \mathrm{P}_{m^0}^\top \boldsymbol{\eta}_A \\
&= \mathrm{S}_{xt,m^+}^\top \left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) - \mathrm{S}_{xt,m^+}^\top \Lambda_t \mathrm{S}_{m^0} \left(\mathrm{S}_{m^0}^\top \Lambda_t \mathrm{S}_{m^0}\right)^{-1} \mathrm{S}_{m^0}^\top \left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) \\
\Lambda_B &= \mathrm{P}_{xt,m^+}^\top \Lambda_A \mathrm{P}_{xt,m^+} - \mathrm{P}_{xt,m^+}^\top \Lambda_A \mathrm{P}_{m^0} \left(\mathrm{P}_{m^0}^\top \Lambda_A \mathrm{P}_{m^0}\right)^{-1} \mathrm{P}_{m^0}^\top \Lambda_A \mathrm{P}_{xt,m^+} \\
&= \mathrm{S}_{xt,m^+}^\top \Lambda_t \mathrm{S}_{xt,m^+} - \mathrm{S}_{xt,m^+}^\top \Lambda_t \mathrm{S}_{m^0} \left(\mathrm{S}_{m^0}^\top \Lambda_t \mathrm{S}_{m^0}\right)^{-1} \mathrm{S}_{m^0}^\top \Lambda_t \mathrm{S}_{xt,m^+}
\end{aligned} \tag{B.13}$$

B.2.3 Calculation of $p_C(\mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})$

This distribution is obtained by marginalizing out both the deactive features, $\mathbf{m}^0$, and the robot state, $\mathbf{x}_t$, from p_A:

$$\begin{aligned}
p_C(\mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha}) &= \iint p_A(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha}) d\mathbf{x}_t d\mathbf{m}^0 \\
&= \mathcal{N}(\mathrm{S}_{m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\mu}_C, \Sigma_C) = \mathcal{N}^{-1}(\mathrm{S}_{m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\eta}_C, \Lambda_C).
\end{aligned}$$

Covariance Form

$$\begin{aligned}
\boldsymbol{\mu}_C &= \mathrm{P}_{m^+}^\top \boldsymbol{\mu}_A \\
&= \mathrm{S}_{m^+}^\top \left(\boldsymbol{\mu}_t + \boldsymbol{\mu}_\alpha\right) \\
\Sigma_C &= \mathrm{P}_{m^+}^\top \Sigma_A \mathrm{P}_{m^+} \\
&= \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^-} \left(\mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^-}\right)^{-1} \mathrm{S}_{m^-}^\top \Sigma_t \mathrm{S}_{m^+}
\end{aligned} \tag{B.14}$$

Information Form

$$\begin{aligned}
\boldsymbol{\eta}_C &= \mathrm{P}_{m^+}^\top \boldsymbol{\eta}_A - \mathrm{P}_{m^+}^\top \Lambda_A \mathrm{P}_{xt,m^0} \left(\mathrm{P}_{xt,m^0}^\top \Lambda_A \mathrm{P}_{xt,m^0}\right)^{-1} \mathrm{P}_{xt,m^0}^\top \boldsymbol{\eta}_A \\
&= \mathrm{S}_{m^+}^\top \left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) - \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{xt,m^0} \left(\mathrm{S}_{xt,m^0}^\top \Lambda_t \mathrm{S}_{xt,m^0}\right)^{-1} \mathrm{S}_{xt,m^0}^\top \left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\alpha\right) \\
\Lambda_C &= \mathrm{P}_{m^+}^\top \Lambda_A \mathrm{P}_{m^+} - \mathrm{P}_{m^+}^\top \Lambda_A \mathrm{P}_{xt,m^0} \left(\mathrm{P}_{xt,m^0}^\top \Lambda_A \mathrm{P}_{xt,m^0}\right)^{-1} \mathrm{P}_{xt,m^0}^\top \Lambda_A \mathrm{P}_{m^+} \\
&= \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{xt,m^0} \left(\mathrm{S}_{xt,m^0}^\top \Lambda_t \mathrm{S}_{xt,m^0}\right)^{-1} \mathrm{S}_{xt,m^0}^\top \Lambda_t \mathrm{S}_{m^+}
\end{aligned} \tag{B.15}$$

B.2.4 Calculation of $p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$

This distribution is obtained by marginalizing out the robot state, $\mathbf{x}_t$, from our original posterior $p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$:

$$\begin{aligned} p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= \int p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) d\mathbf{x}_t \\ &= \mathcal{N}(\mathrm{S}_{m^0,m^+,m^-}^\top \boldsymbol{\xi}_t; \boldsymbol{\mu}_D, \Sigma_D) = \mathcal{N}^{-1}(\mathrm{S}_{m^0,m^+,m^-}^\top \boldsymbol{\xi}_t; \boldsymbol{\eta}_D, \Lambda_D). \end{aligned}$$

Covariance Form

$$\begin{aligned} \boldsymbol{\mu}_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \boldsymbol{\mu}_t \\ \Sigma_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \Sigma_t \mathrm{S}_{m^0,m^+,m^-} \end{aligned} \tag{B.16}$$

Information Form

$$\begin{aligned} \boldsymbol{\eta}_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \boldsymbol{\eta}_t - \mathrm{S}_{m^0,m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t} \left(\mathrm{S}_{x_t}^\top \Lambda_t \mathrm{S}_{x_t}\right)^{-1} \mathrm{S}_{x_t}^\top \boldsymbol{\eta}_t \\ \Lambda_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \Lambda_t \mathrm{S}_{m^0,m^+,m^-} - \mathrm{S}_{m^0,m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t} \left(\mathrm{S}_{x_t}^\top \Lambda_t \mathrm{S}_{x_t}\right)^{-1} \mathrm{S}_{x_t}^\top \Lambda_t \mathrm{S}_{m^0,m^+,m^-} \end{aligned} \tag{B.17}$$

B.2.5 Calculation of $\tilde{p}_{\mathbf{SEIF}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$

To calculate the SEIF sparsified posterior we combine the three distributions p_B (B.12)–(B.13), p_C (B.14)–(B.15), and p_D (B.16)–(B.17) according to (B.9):

$$\begin{aligned} \tilde{p}_{\mathrm{SEIF}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= \frac{p_B(\mathbf{x}_t, \mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})}{p_C(\mathbf{m}^+ | \mathbf{m}^- = \boldsymbol{\alpha})} p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \\ &= \mathcal{N}(\boldsymbol{\xi}_t; \tilde{\boldsymbol{\mu}}_t, \tilde{\Sigma}_t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \tilde{\boldsymbol{\eta}}_t, \tilde{\Lambda}_t). \end{aligned}$$

Covariance Form

$$
\begin{aligned}
&\tilde{p}_{\mathrm{SEIF}}\big(\mathbf{x}_t,\mathbf{m}^0,\mathbf{m}^+,\mathbf{m}^-\big)\\
&\quad\propto \exp\Big\{-\tfrac{1}{2}\big(\mathrm{S}_{x_t,m^+}^\top\boldsymbol{\xi}_t-\boldsymbol{\mu}_B\big)^\top\Sigma_B^{-1}\big(\mathrm{S}_{x_t,m^+}^\top\boldsymbol{\xi}_t-\boldsymbol{\mu}_B\big)\\
&\qquad\quad+\tfrac{1}{2}\big(\mathrm{S}_{m^+}^\top\boldsymbol{\xi}_t-\boldsymbol{\mu}_C\big)^\top\Sigma_C^{-1}\big(\mathrm{S}_{m^+}^\top\boldsymbol{\xi}_t-\boldsymbol{\mu}_C\big)\\
&\qquad\quad-\tfrac{1}{2}\big(\mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\xi}_t-\boldsymbol{\mu}_D\big)^\top\Sigma_D^{-1}\big(\mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\xi}_t-\boldsymbol{\mu}_D\big)\Big\}\\
&\quad=\exp\Big\{-\tfrac{1}{2}\big(\boldsymbol{\xi}_t-(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\big)^\top\mathrm{S}_{x_t,m^+}\Sigma_B^{-1}\mathrm{S}_{x_t,m^+}^\top\big(\boldsymbol{\xi}_t-(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\big)\\
&\qquad\quad+\tfrac{1}{2}\big(\boldsymbol{\xi}_t-(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\big)^\top\mathrm{S}_{m^+}\Sigma_C^{-1}\mathrm{S}_{m^+}^\top\big(\boldsymbol{\xi}_t-(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\big)\\
&\qquad\quad-\tfrac{1}{2}\big(\boldsymbol{\xi}_t-\boldsymbol{\mu}_t\big)^\top\mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^\top\big(\boldsymbol{\xi}_t-\boldsymbol{\mu}_t\big)\Big\}
\end{aligned}
$$

defining $\Sigma_E^{-1}=\mathrm{S}_{x_t,m^+}\Sigma_B^{-1}\mathrm{S}_{x_t,m^+}^\top-\mathrm{S}_{m^+}\Sigma_C^{-1}\mathrm{S}_{m^+}^\top$ and $\Sigma_F^{-1}=\mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^\top$

$$
\begin{aligned}
&=\exp\Big\{-\tfrac{1}{2}\big(\boldsymbol{\xi}_t-(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\big)^\top\Sigma_E^{-1}\big(\boldsymbol{\xi}_t-(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\big)\\
&\qquad\qquad-\tfrac{1}{2}\big(\boldsymbol{\xi}_t-\boldsymbol{\mu}_t\big)^\top\Sigma_F^{-1}\big(\boldsymbol{\xi}_t-\boldsymbol{\mu}_t\big)\Big\}\\
&=\exp\Big\{-\tfrac{1}{2}\Big(\boldsymbol{\xi}_t^\top\Sigma_E^{-1}\boldsymbol{\xi}_t-2\boldsymbol{\xi}_t^\top\Sigma_E^{-1}(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)+(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)^\top\Sigma_E^{-1}(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)\\
&\qquad\qquad+\boldsymbol{\xi}_t^\top\Sigma_F^{-1}\boldsymbol{\xi}_t-2\boldsymbol{\xi}_t^\top\Sigma_F^{-1}\boldsymbol{\mu}_t+\boldsymbol{\mu}_t^\top\Sigma_F^{-1}\boldsymbol{\mu}_t\Big)\Big\}\\
&=\exp\Big\{-\tfrac{1}{2}\Big(\boldsymbol{\xi}_t^\top(\Sigma_E^{-1}+\Sigma_F^{-1})\boldsymbol{\xi}_t-2\boldsymbol{\xi}_t^\top\Big[\Sigma_E^{-1}(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)+\Sigma_F^{-1}\boldsymbol{\mu}_t\Big]\\
&\qquad\qquad+\Big[(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)^\top\Sigma_E^{-1}(\boldsymbol{\mu}_t+\boldsymbol{\mu}_\alpha)+\boldsymbol{\mu}_t^\top\Sigma_D^{-1}\boldsymbol{\mu}_t\Big]\Big)\Big\}\\
&=\exp\Big\{-\tfrac{1}{2}\Big(\boldsymbol{\xi}_t-(\Sigma_E^{-1}+\Sigma_F^{-1})^{-1}\Big[(\Sigma_E^{-1}+\Sigma_F^{-1})\boldsymbol{\mu}_t+\Sigma_E^{-1}\boldsymbol{\mu}_\alpha\Big]\Big)^\top\times\\
&\qquad\qquad(\Sigma_E^{-1}+\Sigma_F^{-1})\Big(\boldsymbol{\xi}_t-(\Sigma_E^{-1}+\Sigma_F^{-1})^{-1}\Big[(\Sigma_E^{-1}+\Sigma_F^{-1})\boldsymbol{\mu}_t+\Sigma_E^{-1}\boldsymbol{\mu}_\alpha\Big]\Big)\Big\}\\
&=\exp\Big\{-\tfrac{1}{2}\Big(\boldsymbol{\xi}_t-\Big[\boldsymbol{\mu}_t+(\Sigma_E^{-1}+\Sigma_F^{-1})^{-1}\Sigma_E^{-1}\boldsymbol{\mu}_\alpha\Big]\Big)^\top\times\\
&\qquad\qquad(\Sigma_E^{-1}+\Sigma_F^{-1})\Big(\boldsymbol{\xi}_t-\Big[\boldsymbol{\mu}_t+(\Sigma_E^{-1}+\Sigma_F^{-1})^{-1}\Sigma_E^{-1}\boldsymbol{\mu}_\alpha\Big]\Big)\Big\}\\
&=\exp\Big\{-\tfrac{1}{2}\big(\boldsymbol{\xi}_t-\tilde{\boldsymbol{\mu}}_t\big)^\top\tilde{\Sigma}_t^{-1}\big(\boldsymbol{\xi}_t-\tilde{\boldsymbol{\mu}}_t\big)\Big\}
\end{aligned}
$$

where

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_t + \left(\Sigma_E^{-1} + \Sigma_F^{-1}\right)^{-1}\Sigma_E^{-1}\boldsymbol{\mu}_\alpha \\ &= \boldsymbol{\mu}_t + \tilde{\Sigma}_t\left(\mathrm{S}_{x_t,m^+}\Sigma_B^{-1}\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Sigma_C^{-1}\mathrm{S}_{m^+}^\top\right)\boldsymbol{\mu}_\alpha \\ \tilde{\Sigma}_t &= \left(\Sigma_E^{-1} + \Sigma_F^{-1}\right)^{-1} \\ &= \left(\mathrm{S}_{x_t,m^+}\Sigma_B^{-1}\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Sigma_C^{-1}\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^\top\right)^{-1}\end{aligned} \tag{B.18}$$

Information Form

$$\begin{aligned}\tilde{p}_{\text{SEIF}}&\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \\ &\propto \exp\Big\{-\tfrac{1}{2}\left(\mathrm{S}_{x_t,m^+}^\top\boldsymbol{\xi}_t\right)^\top\Lambda_B\left(\mathrm{S}_{x_t,m^+}^\top\boldsymbol{\xi}_t\right) + \boldsymbol{\eta}_B^\top\left(\mathrm{S}_{x_t,m^+}^\top\boldsymbol{\xi}_t\right) \\ &\qquad +\tfrac{1}{2}\left(\mathrm{S}_{m^+}^\top\boldsymbol{\xi}_t\right)^\top\Lambda_C\left(\mathrm{S}_{m^+}^\top\boldsymbol{\xi}_t\right) - \boldsymbol{\eta}_C^\top\left(\mathrm{S}_{m^+}^\top\boldsymbol{\xi}_t\right) \\ &\qquad -\tfrac{1}{2}\left(\mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\xi}_t\right)^\top\Lambda_D\left(\mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\xi}_t\right) + \boldsymbol{\eta}_D^\top\left(\mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\xi}_t\right)\Big\} \\ &= \exp\Big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top\mathrm{S}_{x_t,m^+}\Lambda_B\mathrm{S}_{x_t,m^+}^\top\boldsymbol{\xi}_t + \left(\mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_B\right)^\top\boldsymbol{\xi}_t \\ &\qquad +\tfrac{1}{2}\boldsymbol{\xi}_t^\top\mathrm{S}_{m^+}\Lambda_C\mathrm{S}_{m^+}^\top\boldsymbol{\xi}_t - \left(\mathrm{S}_{m^+}\boldsymbol{\eta}_C\right)^\top\boldsymbol{\xi}_t \\ &\qquad -\tfrac{1}{2}\boldsymbol{\xi}_t^\top\mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\xi}_t + \left(\mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D\right)^\top\boldsymbol{\xi}_t\Big\} \\ &= \exp\Big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top\tilde{\Lambda}_t\boldsymbol{\xi}_t^\top + \tilde{\boldsymbol{\eta}}_t^\top\boldsymbol{\xi}_t\Big\}\end{aligned}$$

where

$$\begin{aligned}\tilde{\boldsymbol{\eta}}_t &= \mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_B - \mathrm{S}_{m^+}\boldsymbol{\eta}_C + \mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D \\ \tilde{\Lambda}_t &= \mathrm{S}_{x_t,m^+}\Lambda_B\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Lambda_C\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}_{m^0,m^+,m^-}^\top\end{aligned} \tag{B.19}$$

B.3 Modified Sparsification Rule

We begin by writing the sparsified posterior approximation as a ratio of the three individual distributions p_U, p_V, p_D as described in §3.3.2:

$$\breve{p}_{\text{ModRule}}\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) = \frac{p_U\left(\mathbf{x}_t, \mathbf{m}^+\right)}{p_V\left(\mathbf{m}^+\right)} p_D\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right). \tag{B.20}$$

Next, we calculate the individual terms of (B.20) by marginalizing and conditioning over our original distribution $p\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) = \mathcal{N}\left(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t\right) = \mathcal{N}^{-1}\left(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t\right)$.

B.3.1 Calculation of $p_U(\mathbf{x}_t, \mathbf{m}^+)$

This distribution is obtained by marginalizing out both the deactivated features, $\mathbf{m}^0$, and the passive features, $\mathbf{m}^-$, from our original posterior $p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$:

$$\begin{aligned} p_U(\mathbf{x}_t, \mathbf{m}^+) &= \iint p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) d\mathbf{m}^0 d\mathbf{m}^- \\ &= \mathcal{N}(\mathrm{S}_{x_t,m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\mu}_U, \Sigma_U) = \mathcal{N}^{-1}(\mathrm{S}_{x_t,m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\eta}_U, \Lambda_U). \end{aligned}$$

Covariance Form

$$\begin{aligned} \boldsymbol{\mu}_U &= \mathrm{S}_{x_t,m^+}^\top \boldsymbol{\mu}_t \\ \Sigma_U &= \mathrm{S}_{x_t,m^+}^\top \Sigma_t \mathrm{S}_{x_t,m^+} \end{aligned} \tag{B.21}$$

Information Form

$$\begin{aligned} \boldsymbol{\eta}_U &= \mathrm{S}_{x_t,m^+}^\top \boldsymbol{\eta}_t - \mathrm{S}_{x_t,m^+}^\top \Lambda_t \mathrm{S}_{m^0,m^-} \left(\mathrm{S}_{m^0,m^-}^\top \Lambda_t \mathrm{S}_{m^0,m^-}\right)^{-1} \mathrm{S}_{m^0,m^-}^\top \boldsymbol{\eta}_t \\ \Lambda_U &= \mathrm{S}_{x_t,m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^+} - \mathrm{S}_{x_t,m^+}^\top \Lambda_t \mathrm{S}_{m^0,m^-} \left(\mathrm{S}_{m^0,m^-}^\top \Lambda_t \mathrm{S}_{m^0,m^-}\right)^{-1} \mathrm{S}_{m^0,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^+} \end{aligned} \tag{B.22}$$

B.3.2 Calculation of $p_V(\mathbf{m}^+)$

This distribution is obtained by marginalizing out all terms except for the active features, $\mathbf{m}^+$, from our original posterior $p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$:

$$\begin{aligned} p_V(\mathbf{m}^+) &= \iiint p(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) d\mathbf{x}_t d\mathbf{m}^0 d\mathbf{m}^- \\ &= \mathcal{N}(\mathrm{S}_{m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\mu}_V, \Sigma_V) = \mathcal{N}^{-1}(\mathrm{S}_{m^+}^\top \boldsymbol{\xi}_t; \boldsymbol{\eta}_V, \Lambda_V). \end{aligned}$$

Covariance Form

$$\begin{aligned} \boldsymbol{\mu}_V &= \mathrm{S}_{m^+}^\top \boldsymbol{\mu}_t \\ \Sigma_V &= \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^+} \end{aligned} \tag{B.23}$$

Information Form

$$\begin{aligned} \boldsymbol{\eta}_V &= \mathrm{S}_{m^+}^\top \boldsymbol{\eta}_t - \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-} \left(\mathrm{S}_{x_t,m^0,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-}\right)^{-1} \mathrm{S}_{x_t,m^0,m^-}^\top \boldsymbol{\eta}_t \\ \Lambda_V &= \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-} \left(\mathrm{S}_{x_t,m^0,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-}\right)^{-1} \mathrm{S}_{x_t,m^0,m^-}^\top \Lambda_t \mathrm{S}_{m^+} \end{aligned} \tag{B.24}$$

B.3.3 Calculation of $\breve{p}_{\mathbf{ModRule}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-)$

To calculate the Modified-Rule sparsified posterior we combine the three distributions p_U (B.21)–(B.22), p_V (B.23)–(B.24), and p_D (B.16)–(B.17) according to (B.20):

$$\begin{aligned} \breve{p}_{\textsc{ModRule}}(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) &= \frac{p_U(\mathbf{x}_t, \mathbf{m}^+)}{p_V(\mathbf{m}^+)} p_D(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-) \\ &= \mathcal{N}(\boldsymbol{\xi}_t; \breve{\boldsymbol{\mu}}_t, \breve{\Sigma}_t) = \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \breve{\boldsymbol{\eta}}_t, \breve{\Lambda}_t). \end{aligned}$$

Covariance Form

$$\begin{aligned}
&\breve{p}_{\textsc{ModRule}}\big(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big)\\
&\qquad\propto \exp\Big\{-\tfrac{1}{2}\big(\mathrm{S}_{x_t,m^+}^{\top}\boldsymbol{\xi}_t - \boldsymbol{\mu}_U\big)^{\top}\Sigma_U^{-1}\big(\mathrm{S}_{x_t,m^+}^{\top}\boldsymbol{\xi}_t - \boldsymbol{\mu}_U\big)\\
&\qquad\qquad +\tfrac{1}{2}\big(\mathrm{S}_{m^+}^{\top}\boldsymbol{\xi}_t - \boldsymbol{\mu}_V\big)^{\top}\Sigma_V^{-1}\big(\mathrm{S}_{m^+}^{\top}\boldsymbol{\xi}_t - \boldsymbol{\mu}_V\big)\\
&\qquad\qquad -\tfrac{1}{2}\big(\mathrm{S}_{m^0,m^+,m^-}^{\top}\boldsymbol{\xi}_t - \boldsymbol{\mu}_W\big)^{\top}\Sigma_W^{-1}\big(\mathrm{S}_{m^0,m^+,m^-}^{\top}\boldsymbol{\xi}_t - \boldsymbol{\mu}_W\big)\Big\}\\
&\qquad= \exp\Big\{-\tfrac{1}{2}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big)^{\top}\mathrm{S}_{x_t,m^+}\Sigma_U^{-1}\mathrm{S}_{x_t,m^+}^{\top}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big)\\
&\qquad\qquad +\tfrac{1}{2}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big)^{\top}\mathrm{S}_{m^+}\Sigma_V^{-1}\mathrm{S}_{m^+}^{\top}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big)\\
&\qquad\qquad -\tfrac{1}{2}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big)^{\top}\mathrm{S}_{m^0,m^+,m^-}\Sigma_W^{-1}\mathrm{S}_{m^0,m^+,m^-}^{\top}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big)\Big\}\\
&\qquad= \exp\Big\{-\tfrac{1}{2}\big(\boldsymbol{\xi}_t - \breve{\boldsymbol{\mu}}_t\big)^{\top}\breve{\Sigma}_t^{-1}\big(\boldsymbol{\xi}_t - \breve{\boldsymbol{\mu}}_t\big)\Big\}
\end{aligned}$$

where

$$\begin{aligned}
\breve{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_t\\
\breve{\Sigma}_t &= \big(\mathrm{S}_{x_t,m^+}\Sigma_U^{-1}\mathrm{S}_{x_t,m^+}^{\top} - \mathrm{S}_{m^+}\Sigma_V^{-1}\mathrm{S}_{m^+}^{\top} + \mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^{\top}\big)^{-1}
\end{aligned} \tag{B.25}$$

Information Form

$$\begin{aligned}
&\breve{p}_{\textsc{ModRule}}\big(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\big)\\
&\qquad\propto \exp\Big\{-\tfrac{1}{2}\big(\mathrm{S}_{x_t,m^+}^{\top}\boldsymbol{\xi}_t\big)^{\top}\Lambda_U\big(\mathrm{S}_{x_t,m^+}^{\top}\boldsymbol{\xi}_t\big) + \boldsymbol{\eta}_U^{\top}\big(\mathrm{S}_{x_t,m^+}^{\top}\boldsymbol{\xi}_t\big)\\
&\qquad\qquad +\tfrac{1}{2}\big(\mathrm{S}_{m^+}^{\top}\boldsymbol{\xi}_t\big)^{\top}\Lambda_V\big(\mathrm{S}_{m^+}^{\top}\boldsymbol{\xi}_t\big) - \boldsymbol{\eta}_V^{\top}\big(\mathrm{S}_{m^+}^{\top}\boldsymbol{\xi}_t\big)\\
&\qquad\qquad -\tfrac{1}{2}\big(\mathrm{S}_{m^0,m^+,m^-}^{\top}\boldsymbol{\xi}_t\big)^{\top}\Lambda_W\big(\mathrm{S}_{m^0,m^+,m^-}^{\top}\boldsymbol{\xi}_t\big) + \boldsymbol{\eta}_W^{\top}\big(\mathrm{S}_{m^0,+,m^-}^{\top}\boldsymbol{\xi}_t\big)\Big\}\\
&\qquad= \exp\Big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^{\top}\mathrm{S}_{x_t,m^+}\Lambda_U\mathrm{S}_{x_t,m^+}^{\top}\boldsymbol{\xi}_t + \big(\mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_U\big)^{\top}\boldsymbol{\xi}_t\\
&\qquad\qquad +\tfrac{1}{2}\boldsymbol{\xi}_t^{\top}\mathrm{S}_{m^+}\Lambda_V\mathrm{S}_{m^+}^{\top}\boldsymbol{\xi}_t - \big(\mathrm{S}_{m^+}\boldsymbol{\eta}_V\big)^{\top}\boldsymbol{\xi}_t\\
&\qquad\qquad -\tfrac{1}{2}\boldsymbol{\xi}_t^{\top}\mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}_{m^0,m^+,m^-}^{\top}\boldsymbol{\xi}_t + \big(\mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D\big)^{\top}\boldsymbol{\xi}_t\Big\}\\
&\qquad= \exp\Big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^{\top}\breve{\Lambda}_t\boldsymbol{\xi}_t^{\top} + \breve{\boldsymbol{\eta}}_t^{\top}\boldsymbol{\xi}_t\Big\}
\end{aligned}$$

where

$$\begin{aligned}\breve{\boldsymbol{\eta}}_t &= \mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_U - \mathrm{S}_{m^+}\boldsymbol{\eta}_V + \mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D \\ \breve{\Lambda}_t &= \mathrm{S}_{x_t,m^+}\Lambda_U\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Lambda_V\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}_{m^0,m^+,m^-}^\top\end{aligned} \tag{B.26}$$

BIBLIOGRAPHY

[1] F. Aguirre, J.M. Boucher, and J.J. Jacq. Underwater navigation by video sequence analysis. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 537–539, Atlantic City, NJ, USA, June 1990.

[2] E.A. Allen. Research submarine ALVIN. In *Proceedings*, pages 138–140. U.S. Naval Institute, 1964.

[3] F. Badra, A. Qumsieh, and G. Dudek. Rotation and zooming in image mosaicing. In *Proceedings of Fourth IEEE Workshop on Applications of Computer Vision*, pages 50–55, Princeton, NJ, October 1998.

[4] R.D. Ballard. Hydrothermal vent fields of the East Pacific Rise at 21 deg.N, and Galapagos Rift at 86 deg.W. *EOS, Transactions of the American Geophysical Union*, 60(46), 1979.

[5] R.D. Ballard, L.E. Stager, D. Master, D.R. Yoerger, D.A. Mindell, L.L. Whitcomb, H. Singh, and D. Piechota. Iron age shipwrecks in deep water off Ashkelon, Israel. *American Journal of Archeology*, 106(2), April 2002.

[6] R.D. Ballard, D.R. Yoerger, W.K. Stewart, and A. Bowen. ARGO/JASON: A remotely operated survey and sampling system for full-ocean depth. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, pages 71–75, 1991.

[7] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, Inc., New York, 2001.

[8] P.A. Beardsley, A. Zisserman, and D. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, June 1997.

[9] J.R. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, pages 237–252, 1992.

[10] M. Bosse. *Atlas, a Framework for Scalable Mapping.* PhD thesis, Massachusetts Institute of Technology, 2004.

[11] M. Bosse, P. Newman, J.J. Leonard, and S. Teller. An Atlas framework for scalable mapping. *International Journal of Robotics Research*, 23:1113–1139, December 2004.

[12] A.M. Bradley, M.D. Feezor, H. Singh, and F.Y. Sorrell. Power systems for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 26(4):526–538, October 2001.

[13] A.M Bradley, D.R. Yoerger, B.B. Walden, and H. Singh. The Autonomous Benthic Explorer, an instrument for deep ocean survey. *EOS, Transactions of the American Geophysical Union*, 77(3, Suppl.), February 1996.

[14] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:333–390, 1977.

[15] A.S. Brierley, P.G. Fernandes, M.A. Brandon, F. Armstrong, N.W. Millard, S.D. McPhail, P. Stevenson, M. Pebody, J. Perrett, M. Squires, D.G. Bone, and G. Griffiths. Antarctic krill under sea ice: Elevated abundance in a narrow band just south of ice edge. *Science*, 295:1890–1892, March 2002.

[16] N.A. Brokloff. Matrix algorithm for doppler sonar navigation. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 3, pages 378–383, Brest, France, September 1994.

[17] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[18] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, New York, NY, 3 edition, 1997.

[19] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann. Scalable coordination for wireless sensor networks: Self-configuring localization systems. In *Proc. International Symposium on Communication Theory and Applications*, Ambleside, UK, July 2001.

[20] A. Can and H. Singh. Methods for correcting lighting pattern and attenuation in underwater imagery. *IEEE Journal of Oceanic Engineering*, Submitted, Under Review.

[21] A. Can, C.V. Stewart, B. Roysam, and H.L. Tanenbaum. A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: Application to mosaicing the curved human retina. In *Proceedings of the IEEE Conference on Computer Vision*, volume 2, pages 585–591, Hilton Head Island, SC, USA, June 2000.

[22] J.F. Canny. Finding edges and lines in images. Master's thesis, Massachusetts Institute of Technology, 1983.

[23] Covariance Intersection Working Group (CIWG). A culminating advance in the theory and practice of data fusion, filtering, and decentralized estimation. Technical report, CIWG, 1997.

[24] D. Coleman, R.D. Ballard, and T. Gregory. Marine archaeological exploration of the Black Sea. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 3, pages 1287–1291, September 2003.

[25] J.B. Corliss et al. Submarine thermal springs on the Galapagos Rift. *Science*, 203:1073–1083, 1979.

[26] J.A. Crisp, M. Adler, J.R. Matijevic, S.W. Squyres, R.E. Arvidson, and D.M. Kass. Mars exploration rover mission. *Journal of Geophysical Research*, 108(E12):ROV 2–1, 2003.

[27] A.J. Davison. *Mobile Robot Navigation Using Active Vision*. PhD thesis, University of Oxford, 1999.

[28] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision*, pages 1403–1410, 2003.

[29] U.R. Dhond and J.K. Aggarwal. Structure from stereo — a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510, November/December 1989.

[30] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.

[31] J.D. Donnelly. 1967 — ALVIN's year of science. *Naval Research Reviews*, 21(1):18–26, 1968.

[32] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer-Verlag, New York, 2001.

[33] T. Duckett, S. Marsland, and J. Shapiro. Learning globally consistent maps by relaxation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3841–3846, San Francisco, CA, April 2000.

[34] S.Q. Duntley. Light in the sea. *Journal of the Optical Society of America*, 53(2):214–233, February 1963.

[35] B. Elder, A.D. Bowen, M. Heintz, M. Naiman, C. Taylor, W. Seller, J.C. Howland, and L.L. Whitcomb. Jason 2: A review of capabilities. *EOS, Transactions of the American Geophysical Union 2003 Fall Meeting*, 2003. Abstract.

[36] R. Eustice, O. Pizarro, and H. Singh. Visually augmented navigation in an unstructured environment using a delayed state history. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 25–32, New Orleans, USA, April 2004.

[37] R. Eustice, O. Pizarro, H. Singh, and J. Howland. UWIT: Underwater image toolbox for optical image processing and mosaicking in Matlab. In *Proceedings of the International Symposium on Underwater Technology*, pages 141–145, Tokyo, Japan, April 2002.

[38] R. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed state filters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Accepted, To Appear.

[39] R. Eustice, M. Walter, and J. Leonard. Sparse Extended Information Filters: Insights into sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Submitted, Under Review.

[40] O. Faugeras, Q. Luong, and T. Papadopoulu. *The Geometry of Multiple Images.* MIT Press, 2001.

[41] O.D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[42] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communication Association and Computing Machine*, 24(6):381–395, June 1981.

[43] A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proceedings of the European Conference on Computer Vision*, pages 311–326, Freiburg, Germany, June 1998. Springer-Verlag.

[44] S.D. Fleischer. *Bounded-Error Vision-Based Navigation of Autonomous Underwater Vehicles.* PhD thesis, Stanford University, 2000.

[45] S.D. Fleischer, R.L. Marks, S.M. Rock, and M.J. Lee. Improved real-time video mosaicking of the ocean floor. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 3, pages 1935–1944, October 1995.

[46] S.D. Fleischer, S.M. Rock, and R. Burton. Global position determination and vehicle path estimation from a vision sensor for real-time video mosaicking and navigation. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 641–647, October 1997.

[47] B. Fletcher. Chemical plume mapping with an autonomous underwater vehicle. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 508–512, Honolulu, HI, USA, November 2001.

[48] T.I. Fossen. *Guidance and Control of Ocean Vehicles.* John Wiley and Sons Ltd., New York, 1994.

[49] U. Frese. Treemap: An O(Log N) algorithm for simultaneous localization and mapping. In C. Freksa, editor, *Spatial Cognition IV.* Springer Verlag, 2004.

[50] U. Frese. A proof for the approximate sparsity of SLAM information matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 331–337, Barcelona, Spain, 2005.

[51] U. Frese and G. Hirzinger. Simultaneous localization and mapping - a discussion. In *Proceedings of the IJCAI Workshop Reasoning with Uncertainty in Robotics*, pages 17–26, Seattle, WA, 2001.

[52] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):1–12, 2005.

[53] T. Gaiffe. U-Phins: A FOG-based inertial navigation system developed specifically for AUV navigation and control. In *International Conference on Underwater Intervention*, New Orleans, LA, February 2002.

[54] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1982.

[55] N. Gracias and J. Santos-Victor. Trajectory reconstruction using mosaic registration. In *SIRS '99*, Coimbra, Portugal, July 1999.

[56] N. Gracias and J. Santos-Victor. Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding*, 79:66–91, March 2000.

[57] N. Gracias and J. Santos-Victor. Trajectory reconstruction with uncertainty estimation using mosaic registration. In *Robotics and Autonomous Systems*, June 2001.

[58] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Mosaic based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 28(4):609–624, October 2003.

[59] G. Griffiths, N.W. Millard, M. Pebody, and S.D. McPhail. The end of research ships? Autosub - an autonomous underwater vehicle for ocean science. In *Proceedings of Underwater Technology International*, pages 349–362, Aberdeen, April 1997. Society for Underwater Technology.

[60] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

[61] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2000.

[62] P.E. Hagen, N. Størkersen, K. Vestgård, and P. Kartvedt. The HUGIN 1000 autonomous underwater vehicle for military applications. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, pages 22–26, San Diego, CA, USA, September 2003.

[63] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, Manchester, U.K., 1988.

[64] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[65] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision*, pages 1106–1112, Puerto Rico, 1997.

[66] J.C. Hill, N. Driscoll, J. Weissel, M. Kastner, H. Singh, M.H. Cormier, R. Camilli, R. Eustice, R. Lipscomb, N. McPhee, K. Newman, G. Robertson, E. Solomon, and K. Tomanka. A detailed near-bottom survey of large gas blowout structures along the US Atlantic shelf break using the autonomous underwater vehicle (AUV) SeaBED. In *EOS, Transactions of the American Geophysical Union*, Abstract, 2004. In Print.

[67] B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 2nd edition, 1986.

[68] B.K.P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, January 1990.

[69] J. Howland. Digital data logging and processing, Derbyshire survey, 1997. Technical report, Woods Hole Oceanographic Institution, December 1999.

[70] P.C. Hughes. *Spacecraft Attitude Dynamics.* J. Wiley, New York, 1986.

[71] M.M. Hunt, W.M. Marquet, D.A. Moller, K.R. Peal, W.K. Smith, and R.C. Spindel. An acoustic navigation system. Technical Report WHOI-74-6, Woods Hole Oceanographic Institution, December 1974.

[72] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *Proceedings of the International Conference on Computer Vision*, pages 959–966, January 1996.

[73] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proceedings of the International Conference on Computer Vision*, pages 605–611, Cambridge, MA USA, June 1995.

[74] J.S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, April 1990.

[75] S.J. Julier and J.K. Uhlmann. Building a million beacon map. In G.T. McKee and P.S. Schenker, editors, *Sensor Fusion and Decentralized Control in Robotic Systems IV*, volume 4571, pages 10–21. Proc. SPIE, October 2001.

[76] S.B. Kang. A survey of image-based rendering techniques. Technical Report CRL 97/4, Cambridge Research Laboratory, August 1997.

[77] A. Khotanzad and Y.H. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, May 1990.

[78] K. Konolige. Large-scale map-making. In *Proceedings of the AAAI*, pages 457–463, San Jose, CA, 2004.

[79] S. Kruger and A. Calway. A multiresolution frequency domain method for estimating affine motion parameters. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 113–116, Lausanne, Switzerland, September 1996.

[80] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *Proceedings IEEE Workshop on Representation of Visual Scenes*, pages 10–17, Cambridge, MA, June 1995.

[81] R. Kumar, H.S. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P.J. Burt. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, 89(10):1518–1539, October 2001. Invited Paper.

[82] C.H. Langmuir, C. German, P. Michael, D.R. Yoerger, D.J. Fornari, T.M. Shank, P.D. Asimow, and H.N. Edmonds. Hydrothermal prospecting and petrological sampling in the Lau Basin: Background data for the integrated study site. *EOS, Transactions of the American Geophysical Union, Fall Meeting Abstracts*, 85(47):B13A–0189, 2004.

[83] S. Lanser and T. Lengauer. On the selection of candidates for point and line correspondences. In *Proceedings of the International Symposium on Computer Vision*, pages 157–162. IEEE Computer Society Press, 1995.

[84] K.N. Leabourne, S.M. Rock, S.D. Fleischer, and R. Burton. Station keeping of an ROV using vision technology. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 634–640, October 1997.

[85] J.J. Leonard and H.J.S. Feder. Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering*, 26(4):561–571, 2001.

[86] J.J. Leonard and P. Newman. Consistent, convergent, and constant-time SLAM. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 2003.

[87] J.J. Leonard and R.J. Rikoski. Incorporation of delayed decision making into stochastic mapping. In *Experimental Robotics VII*, volume 271 of *Lecture Notes in Control and Information Sciences*, pages 533–542. Springer-Verlag, 2001.

[88] J.J. Leonard, R.J. Rikoski, P.M. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *International Journal of Robotics Research*, 21(10):943–975, October 2002.

[89] S. Lerner, D. Yoerger, and T. Crook. Navigation for the Derbyshire phase 2 survey. Technical report, Woods Hole Oceanographic Institution, 1999.

[90] J. Lim. *Two-Dimensional Signal and Image Processing.* Prentice Hall, Englewood Cliffs, N.J., 1990.

[91] J. Liu, B.C. Vemuri, and J.L. Marroquin. Local frequency representations for robust multimodal image registration. *IEEE Transactions on Medical Imaging*, 21(5):462–469, May 2002.

[92] Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 1227–1233, September 2003.

[93] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[94] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, April 1997.

[95] L. Lucchese. Estimating affine transformations in the frequency domain. In *Proceedings International Conference on Image Processing*, volume 2, pages 909–912, Thessaloniki, Greece, October 2001.

[96] R. Mandelbaum, G. Salgian, and H. Sawhney. Correlation-based estimation of egomotion and structure from motion and stereo. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 544–550, Kerkyra, Greece, September 1999.

[97] R. Mandelbaum, G. Salgian, H.S. Sawhney, and M. Hansen. Terrain reconstruction for ground and underwater robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 879–884, April 2000.

[98] R.L. Marks, S.M. Rock, and M.J. Lee. Real-time video mosaicking of the ocean floor. *IEEE Journal of Oceanic Engineering*, 20(3):229–241, July 1995.

[99] R.L. Marks, H.H. Wang, M.J. Lee, and S.M. Rock. Automatic visual station keeping of an underwater robot. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 2, pages 137–142, Brest, France, September 1994.

[100] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stabe extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[101] R. McEwen, H. Thomas, D. Weber, and F. Psota. Performance of an AUV navigation system at Arctic latitudes. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, pages 642–653, 2003.

[102] B.L. McGlamery. Computer analysis and simulation of underwater camera system performance. Technical Report SIO Ref. 75-2, Scripps Institution of Oceanography, January 1975.

[103] P. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of the International Conference on Computer Vision*, pages 314–320, Boston, MA, 1995.

[104] P.F. McLauchlan. The variable state dimension filter applied to surface-based structure from motion. Technical Report VSSP-TR-4/99, School of Electrical Engineering, Information Technology and Mathematics, University of Surrey, 1999.

[105] P.F. McLauchlan. A batch/recursive algorithm for 3D scene reconstruction. In *Proceedings of the IEEE Conference on Computer Vision*, volume 2, pages 738–743, Hilton Head, SC USA, 2000.

[106] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the European Conference on Computer Vision*, pages 0–7, Copenhagen, Denmark, May 2002.

[107] P.H. Milne. *Underwater Acoustic Positioning Systems.* Gulf Publishing Company, Houston, 1983.

[108] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[109] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.

[110] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *Proceedings of the 1st International Symposium on Experimental Robotics*, Montreal, Canada, June 1989.

[111] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems.* MIT Press, 1999.

[112] S. Negahdaripour. Passive navigation in a planar world. In *International Geo-science and Remote Sensing Symposium*, volume 3, pages 1264–1267, July 1989.

[113] S. Negahdaripour and J. Lanjing. Direct recovery of motion and range from images of scenes with time-varying illumination. In *Proceedings of the International Symposium on Computer Vision*, pages 467–472, Coral Gables, FL, November 1995.

[114] S. Negahdaripour, X. Xu, and L. Jin. Direct estimation of motion from sea floor images for automatic station-keeping of submersible platforms. *IEEE Journal of Oceanic Engineering*, 24(3):370–382, July 1999.

[115] S. Negahdaripour, X. Xu, A. Khamene, and Z. Awan. 3-D motion and depth estimation from sea-floor images for mosaic-based station-keeping and navigation of ROVs/AUVs and high-resolution sea-floor mapping. In *Proceedings of the Workshop on Autonomous Underwater Vehicles*, pages 191–200, Cambridge, MA, USA, August 1998.

[116] S. Negahdaripour and X. Xun. Mosaic-based positioning and improved motion-estimation methods for automatic navigation of submersible vehicles. *IEEE Journal of Oceanic Engineering*, 27(1):79–99, January 2002.

[117] S. Negahdaripour and C.H. Yu. A generalized brightness change model for computing optical flow. In *Proceedings of the International Conference on Computer Vision*, pages 2–11, Berlin, Germany, May 1993.

[118] J. Neira and J.D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, December 2001.

[119] P. V. O'Neil. *Advanced Engineering Mathematics*. Brooks/Cole Publishing Company, Pacific Grove, CA, 4 edition, 1995.

[120] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. Technical Report CSD-02-1198, University of California, Berkeley, September 2002.

[121] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1157–1164, San Francisco, CA, 2003.

[122] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[123] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *Proceedings of the IEEE Conference on Computer Vision*, pages 338–343, June 1997.

[124] O. Pizarro. *Large Scale Structure from Motion for Autonomous Underwater Vehicle Surveys*. PhD thesis, Massachusetts Institute of Technology, September 2004.

[125] O. Pizarro, R. Eustice, and H. Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing Techniques and Applications*, pages 601–612, Sydney, Australia, December 2003.

[126] O. Pizarro and H. Singh. Towards large area mosaicing for underwater scientific applications. *IEEE Journal of Oceanic Engineering*, 28(4):651–672, October 2003.

[127] O. Pizarro, H. Singh, and S. Lerner. Towards image-based characterization of acoustic navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1519–1524, October 2000.

[128] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[129] W.O. Rainnie. The use of the deep submersible ALVIN in oceanography. In *Proceedings of the Governors Conference on Oceanography*, pages 128–143, Rockefeller University, New York, NY, October 1967. State of New York and the New York State Science and Technology Foundation.

[130] RD Instruments. Acoustic Doppler Current Profiler: Principles of operation a practical primer. Technical report, RD Instruments, San Diego, CA, USA, 1996.

[131] RD Instruments. ADCP coordinate transformation. Technical report, RD Instruments, San Diego, CA, USA, 1998.

[132] B.S. Reddy and B.N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, August 1996.

[133] J.R. Reynolds, R.C. Highsmith, B. Konar, C.G. Wheat, and D. Doudna. Fisheries and fisheries habitat investigations using undersea technology. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 2, pages 812–820, Honolulu, HI, USA, November 2001.

[134] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987.

[135] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2003.

[136] H.S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D & 3D dominant motion estimation for mosaicing and video representation. In *Proceedings of the International Conference on Computer Vision*, pages 583–590, Cambridge, MA, June 1995.

[137] H.S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 103–119, Freiburg, Germany, 1998.

[138] H.S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, and V. Paragano. Videobrush: Experiences with consumer video mosaicing. In *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision*, pages 56–62, Princeton, NJ, USA, October 1998.

[139] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proceedings of the International Conference on Computer Vision*, pages 636–643, Vancouver, BC, July 2001.

[140] C. Schmid and R. Mohr. Matching by local invariants. Technical Report 2644, INRIA, August 1995.

[141] T. Shank, D. Fornari, D. Yoerger, S. Humphris, A. Bradley, S. Hammond, J. Lupton, D. Scheirer, R. Collier, A.L. Reysenbach, K. Ding, W. Seyfreid, D. Butterfield, E. Olson, M. Lilley, N. Ward, and J. Eisen. Deep submergence synergy: Alvin and ABE explore the Galapagos rift at 86°w. *EOS, Transactions of the American Geophysical Union*, 84(41):425,432–433, October 2003.

[142] X. Shen, P. Palmer, P. McLauchlan, and A. Hilton. Error propogation from camera motion to epipolar constraint. In *Proceedings of the British Machine Vision Conference*, pages 546–555, September 2000.

[143] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, Carnegie Mellon University, August 1994.

[144] H. Singh, J. Adams, D. Mindell, and B. Foley. Imaging underwater for archeology. *American Journal of Field Archaeology*, 27(3):319–328, 2000.

[145] H. Singh, R. Armstrong, F. Gilbes, R. Eustice, C. Roman, O. Pizarro, and J. Torres. Imaging coral I: Imaging coral habitats with the SeaBED AUV. *The Journal for Subsurface Sensing Technologies and Applications*, 5(1):25–42, January 2004.

[146] H. Singh, A. Can, R. Eustice, S. Lerner, N. McPhee, O. Pizarro, and C. Roman. SeaBED AUV offers new platform for high-resolution imaging. *EOS, Transactions of the American Geophysical Union*, 85(31):289,294–295, August 2004.

[147] H. Singh, R. Eustice, C. Roman, and O. Pizarro. The SeaBED AUV - a platform for high resolution imaging. In *Unmanned Underwater Vehicle Showcase*, Southampton Oceanography Centre, UK, September 2002.

[148] H. Singh and J. Howland. A forensic analysis of the remains of flight EA990. Poster, Woods Hole Oceanographic Institution, 1999.

[149] H. Singh, J. Howland, and O. Pizarro. Advances in large-area photomosaicking underwater. *IEEE Journal of Oceanic Engineering*, 29(3):872–886, July 2004.

[150] H. Singh, F. Weyer, J. Howland, A. Duester, and A.M. Bradley. Quantitative stereo imaging from the Autonomous Benthic Explorer (ABE). In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 52–57, Seattle, WA, September 1999.

[151] H. Singh, L.L. Whitcomb, D.R. Yoerger, and O. Pizarro. Microbathymetric mapping from underwater vehicles in the deep ocean. *Computer Vision and Image Understanding*, 79(1):143–161, July 2000.

[152] H. Singh, D.R. Yoerger, and A.M. Bradley. Issues in design and deployment of AUVs for oceanographic applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1997.

[153] H. Singh, D.R. Yoerger, A.M. Bradley, R. Bachmayer, and W.K. Stewart. Sonar mapping with the Autonomous Benthic Explorer (ABE). In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, September 1995.

[154] R. Smith, M. Self, and P. Cheeseman. *Estimating Uncertain Spatial Relationships in Robotics.* Autonomous Robot Vehicles. Springer-Verlag, 1990.

[155] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In S.S. Lyengar and A. Elfes, editors, *Autonomous Mobile Robots*, pages 323–330. IEEE Computer Society Press, 1991.

[156] G. Strang. *Linear Algebra and Its Applications.* Academic Press, New York, 2nd edition, 1980.

[157] R. Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL 94/2, Cambridge Research Laboratory, May 1994.

[158] J.D. Tardos, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4):311–330, April 2002.

[159] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[160] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H.F. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, July-August 2004.

[161] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 2004. Accepted, To Appear.

[162] T. Tommasini, A. Fusiello, V. Roberto, and E. Trucco. Robust feature tracking in underwater video sequences. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 46–50, September 1998.

[163] B. Triggs, P. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer-Verlag, 2000.

[164] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference*, pages 412–425, 2000.

[165] K. Vestgard, N. Storkesen, and J. Sortland. Seabed surveys with the HUGIN AUV. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, August 1999.

[166] P. Viola and W.M.I. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[167] M.R. Walter and J.J. Leonard. An experimental investigation of cooperative SLAM. In *Proceedings of IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 2004.

[168] Y. Weiss and W.T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[169] L.L. Whitcomb, D.R. Yoerger, and H. Singh. Towards precision robotic maneuvering, survey and manipulation in unstructured undersea environments. In *Proceedings of*

the International Symposium on Robotics Research, pages 45–54, Springer Verlag, London, 1998.

[170] L.L. Whitcomb, D.R. Yoerger, and H. Singh. Advances in Doppler-based navigation of underwater robotic vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 399–406, 1999.

[171] L.L. Whitcomb, D.R. Yoerger, and H. Singh. Combined Doppler/LBL based navigation of underwater vehicles. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, Durham, New Hampshire, May 1999.

[172] L.L. Whitcomb, D.R. Yoerger, H. Singh, and J. Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control and survery operations. In *Proceedings of the International Symposium on Robotics Research*, Springer-Verlag, London, 2000.

[173] S.B. Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems.* PhD thesis, University of Sydney, 2001.

[174] D.R. Yoerger, A.M. Bradley, M.H. Cormier, W.B.F. Ryan, and B.B. Walden. High resolution mapping of a fast spreading mid-ocean ridge with the Autonomous Benthic Explorer. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, August 1999.

[175] D.R. Yoerger, A.M. Bradley, H. Singh, B.B. Walden, M.H. Cormier, and W.B.F. Ryan. Multisensor mapping of the deep seafloor with the Autonomous Benthic Explorer. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, pages 248–253, Tokyo, Japan, May 2000.

[176] D.R. Yoerger, A.M. Bradley, B.B. Walden, M.H. Cormier, and W.B.F. Ryan. Fine-scale seafloor survey in rugged deep-ocean terrain with an autonomous robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1787–1792, San Francisco, CA, USA, April 2000.

[177] D.R. Yoerger and D.A. Mindell. Precise navigation and control of an ROV at 2200 meters depth. In *Proceedings of Intervention/ROV '92*, San Diego, June 1992.

[178] D.R. Yoerger, H. Singh, L.L. Whitcomb, J. Catteau, J. Adams, B. Foley, and D.A. Mindell. High resolution mapping for deep water archeology. In *Annual Meeting of the Society of Historical Archaeology*, 1998.

[179] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–198, 1998.

[180] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

[181] Z. Zhang and Y. Shan. Incremental motion estimation through local bundle adjustment. Technical Report MSR-TR-01-54, Microsoft Research, May 2001.

[182] K. Zuiderveld. Contrast limited adaptive histogram equalization. In Paul Heckbert, editor, *Graphics Gems IV*, volume IV, pages 474–485. Academic Press, Boston, 1994.

50272-101

REPORT DOCUMENTATION PAGE

1. REPORT NO.
MIT/WHOI 2005-08

2.

3. Recipient's Accession No.

4. Title and Subtitle
Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles

5. Report Date
June 2005

6.

7. Author(s)
Ryan M. Eustice

8. Performing Organization Rept. No.

9. Performing Organization Name and Address
MIT/WHOI Joint Program in Oceanography/Applied Ocean Science & Engineering

10. Project/Task/Work Unit No.
MIT/WHOI 2005-08

11. Contract(C) or Grant(G) No.
(C) EEC-9986821
(G)

12. Sponsoring Organization Name and Address
CenSSIS ERC of the National Science Foundation
Woods Hole Oceanographic Institution Penzance Foundation
Department of Defense

13. Type of Report & Period Covered
Ph.D. Thesis

14.

15. Supplementary Notes
This thesis should be cited as: Ryan M. Eustice, 2004. Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles. Ph.D. Thesis. MIT/WHOI, 2005-08.

16. Abstract (Limit: 200 words)

This thesis describes a vision-based, large-area, simultaneous localization and mapping algorithm (SLAM) that respects the low-overlap imagery constraints typical of autonomous underwater vehicles while exploiting the inertial sensor information that is routinely available on such platforms. We adopt a systems-level approach exploiting the complementary aspects of inertial sensing and visual perception from a calibrated pose-instrumented platform. This systems-level strategy yields a robust solution to underwater imaging that overcomes many of the unique challenges of a marine environment (e.g., unstructured terrain, low-overlap imagery, moving light source).

Our large-area SLAM algorithm recursively incorporates relative-pose constraints using a view-based representation that exploits exact sparsity in the Gaussian canonical form. This sparsity allows for efficient O(n) update complexity in the number of images composing the view-based map by utilizing recent multilevel relaxation techniques. We show that our algorithmic formulation is inherently sparse unlike other feature-based canonical SLAM algorithms, which impose sparseness via pruning approximations. In particular, we investigate the sparsification methodology employed by SEIF and offer new insight as to why, and how, its approximation can lead to inconsistencies in the estimated state errors. Lastly, we present a novel algorithm for efficiently extracting consistent marginal covariances useful for data association from the information matrix.

17. Document Analysis a. Descriptors

SLAM
AUVs
computer vision

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement
Approved for publication; distribution unlimited.

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page)

21. No. of Pages
187

22. Price